



UNIVERSITAS INDONESIA

**PERANCANGAN SISTEM KENDALI NONLINEAR DENGAN
MENGUNAKAN MODEL HAMMERSTEIN UNTUK SISTEM
TIGA TANGKI TERHUBUNG**

SKRIPSI

SHIMON KANA B.

0405030729

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

PROGRAM STUDI ELEKTRO

DEPOK

JULI 2009



UNIVERSITAS INDONESIA

**PERANCANGAN SISTEM KENDALI NONLINEAR DENGAN
MENGUNAKAN MODEL HAMMERSTEIN UNTUK SISTEM
TIGA TANGKI TERHUBUNG**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

SHIMON KANA B.

0405030729

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

PROGRAM STUDI ELEKTRO

DEPOK

JULI 2009

PERNYATAAN KEASLIAN KARYA TULIS

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul :

PERANCANGAN SISTEM KENDALI NONLINEAR DENGAN MENGUNAKAN MODEL HAMMERSTEIN UNTUK SISTEM TIGA TANGKI TERHUBUNG

yang dibuat untuk melengkapi persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari skripsi yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesiamaupun Perguruan Tinggi atau instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 17 Juli 2009

Penulis,

Shimon Kana B.

0405030729

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Shimon Kana B.
NPM : 0405030729
Program Studi : Teknik Elektro
Judul Skripsi : Perancangan Sistem Kendali Nonlinear dengan
Menggunakan Model Hammerstein untuk Sistem
Tiga Tangki Terhubung

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Ir. Aries Subiantoro M. SEE ()
Penguji : Dr. Ir. Feri Yusivar M.Eng ()
Penguji : Dr. Abdul Halim M.Eng ()

Ditetapkan di : *WI DEPOK*

Tanggal : *7 JULI 2009*

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Program Studi Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

- (1) Ir. Aries Subiantoro M. SEE., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
- (2) pihak Departemen Elektro Fakultas Teknik Universitas Indonesia yang telah banyak membantu dalam usaha memperoleh data yang saya perlukan;
- (3) orang tua dan keluarga saya yang telah memberikan bantuan dukungan material dan moral; dan
- (4) sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 16 Juli 2009

Penulis

ABSTRAK

Nama : Shimon Kana B.
Program Studi : Teknik Elektro
Judul : Perancangan Sistem Kendali Nonlinear dengan Menggunakan Model Hammerstein untuk Sistem Tiga Tangki Terhubung

Perancangan pengendali untuk sistem nonlinear merupakan suatu persoalan yang cukup rumit. Untuk merealisasikan hal tersebut, dibutuhkan model yang memiliki karakteristik sama seperti proses, yang pada umumnya bersifat nonlinear. Dengan adanya model tersebut, dapat dirancang suatu skema pengendalian *Internal Model Control* (IMC) yang dapat mengendalikan proses yang bersifat nonlinear.

Pada skripsi ini, sebuah model Hammerstein yang terdiri dari blok nonlinear *Neural Network* dan blok linear dengan struktur ARX dirancang untuk meniru karakteristik proses nonlinear yang dimiliki oleh sistem tiga tangki terhubung. Model tersebut diperoleh dengan mengidentifikasi proses melalui pasangan data masukan keluaran sistem lingkaran terbuka. Dengan metode yang sama, invers model tersebut juga dirancang untuk menyusun sebuah skema pengendalian IMC. Selain itu, dirancang juga sebuah blok tambahan *Radial Basis Function Network* (RBFN) untuk meningkatkan performa dari sistem. Kinerja pengendali yang dihasilkan kemudian dibandingkan dengan skema pengendalian PID dan di uji kemampuannya dalam mengatasi terjadinya gangguan berupa kebocoran pada sistem.

Kata kunci: nonlinear, Hammerstein, IMC, *neural network*, ARX, RBFN

ABSTRACT

Name : Shimon Kana B.
Study Program : Electrical Engineering
Title : Design of a Nonlinear Control System using Hammerstein
Model for Three Coupled Tank System

The design of a controller for nonlinear systems is a one complex problem. In order to realize that, it is necessary to obtain a model with the same characteristics with the process, which is in general nonlinear. Using that model, a control scheme of Internal Model Control (IMC) can be designed, which is able to control nonlinear processes.

In this final project, a Hammerstein model which is consist of a nonlinear block of neural network and a linear block of ARX structure is designed to have the same characteristic with nonlinear process of a three coupled tank system. The model is obtained by identifying the process using the input-output data pair of the open loop system. Using the same method, an inverse of the model also design in order to create IMC control scheme. That aside, an additional block of a Radial Basis Function Network (RBFN) also designed to improve the performance of the system. Moreover, the performance of the controller is compared with PID control scheme and its ability to overcome disturbance, system leak, is also tested.

Keywords: nonlinear, Hammerstein, IMC, *neural network*, ARX, RBFN

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN SKRIPSI	ii
LEMBAR PENGESAHAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	v
DAFTAR ISI	vii
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
DAFTAR LAMPIRAN	xii
1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Pembahasan	2
1.3 Pembatasan Masalah	3
1.4 Sistematika Penulisan	3
2. DASAR TEORI	4
2.1 Model Proses Hammerstein	4
2.1.1 Neural Network	5
2.1.2 Fungsi Aktivasi Nonlinear Sigmoid	7
2.1.3 Backpropagation pada Neural Network	8
2.1.4 Sigmoid Network	9
2.1.5 Model Linear ARX	10
2.2 Inverse Model Hammerstein	12
2.3 Radial Basis Function Network (RBFN)	13
2.4 Internal Model Control (IMC)	16
3. PERANCANGAN PENGENDALI INTERNAL MODEL CONTROL UNTUK SISTEM TIGA TANGKI TERHUBUNG	19
3.1 Penurunan Model Nonlinear Sistem Tiga Tangki Terhubung	19
3.2 Identifikasi Model Hammerstein	23
3.2.1 Pengumpulan Data Sistem Tiga Tangki Terhubung	25

3.2.2	Penentuan Waktu Pencuplikan.....	26
3.2.3	Penentuan Parameter-parameter Model Hammerstein	27
3.2.4	Struktur Model Hammerstein.....	28
3.3	Validasi Model Hammerstein.....	31
3.4	Perancangan Invers Model Hammerstein	34
4.	UJI COBA DAN ANALISA.....	38
4.1	Simulasi Pengendali IMC untuk Sistem Tiga Tangki Terhubung.....	38
4.2	Perancangan <i>Radial Basis Function Network</i> (RBFN).....	41
4.3	Perbandingan Kinerja Pengendali IMC dan Skema Pengendali PID	45
4.4	Kinerja Pengendali IMC Terhadap Gangguan	48
	BAB 5 KESIMPULAN	53
	DAFTAR ACUAN	54
	DAFTAR PUSTAKA.....	55
	LAMPIRAN A	56
	LAMPIRAN B	57
	LAMPIRAN C	58
	LAMPIRAN D.....	59

DAFTAR GAMBAR

Gambar 2.1	Struktur model Hammerstein	4
Gambar 2.2	Struktur neuron tunggal	6
Gambar 2.3	Topologi neural network.....	7
Gambar 2.4	Skema open loop control.....	12
Gambar 2.5	<i>Radial Basis Function Network</i>	13
Gambar 2.6	Internal Model Control	16
Gambar 3.1	Sistem tiga tangki terhubung.....	20
Gambar 3.2	Prosedur perancangan pengendali IMC	23
Gambar 3.3	Algoritma identifikasi model Hammerstein.....	24
Gambar 3.4	Gambar masukan dan keluaran untuk identifikasi sistem.....	25
Gambar 3.5	Diagram blok pengambilan pasangan data training	26
Gambar 3.6	Respons tanggapan sistem lingkaran terbuka	27
Gambar 3.7	Pergerakan <i>cost function</i> terhadap jumlah iterasi.....	30
Gambar 3.8	Validasi model dengan masukan sinyal multifrekuensi	32
Gambar 3.9	Kesalahan pada validasi sinyal multifrekuensi	33
Gambar 3.10	Validasi dengan sinyal masukan fungsi step.....	33
Gambar 3.11	Pergerakan <i>cost function</i> terhadap jumlah iterasi pada penurunan invers model	37
Gambar 4.1	Diagram blok yang digunakan pada simulasi pengendali IMC	38
Gambar 4.2	Hasil simulasi pengendalian IMC pada sistem tiga tangki terhubung	39

Gambar 4.3	Kesalahan yang terjadi pada simulasi sistem dengan pengendali IMC.....	40
Gambar 4.4	Sinyal kendali IMC yang menjadi masukan sistem tiga tangki terhubung	40
Gambar 4.5	Blok diagram simulasi pengendali IMC dengan RBFN	42
Gambar 4.6	Hasil simulasi pengendali IMC dengan RBFN dan tanpa RBFN ($\alpha=0.002$)	43
Gambar 4.7	Hasil simulasi pengendali IMC dengan RBFN dan tanpa RBFN ($\alpha=0.0027$)	43
Gambar 4.8	Hasil simulasi pengendali IMC dengan RBFN dan tanpa RBFN ($\alpha=0.0035$)	44
Gambar 4.9	Hasil simulasi pengendali IMC dengan RBFN dan tanpa RBFN ($\alpha=0.0045$)	44
Gambar 4.10	Blok diagram simulasi sistem dengan pengendali PID	45
Gambar 4.11	Hasil simulasi perbandingan IMC dan PID	47
Gambar 4.12	Perbandingan sinyal kendali IMC dan PID.....	48
Gambar 4.13	Sistem tiga tangki terhubung dengan kebocoran.....	48
Gambar 4.14	Respons sistem terhadap kebocoran pada tangki 1	49
Gambar 4.15	Respons sistem terhadap kebocoran pada tangki 2	50
Gambar 4.16	Respons sistem terhadap kebocoran pada tangki 3	50
Gambar 4.17	Sinyal kendali untuk sistem dengan kebocoran pada tangki 3.....	51
Gambar 4.18	Respons sistem untuk nilai acuan melebihi tinggi maksimum	52

DAFTAR TABEL

Tabel 3.1	Spesifikasi sistem tiga tangki terhubung	22
Tabel 3.2	Parameter-parameter blok nonlinear Hammerstein	28
Tabel 3.3	Nilai RMSE dan VAF pada proses validasi model.....	34
Tabel 3.4	Parameter-parameter dari blok nonlinear dari invers model Hammerstein.....	35
Tabel 4.1	Titik tengah dan lebar <i>cluster</i> RBFN	41
Tabel 4.2	Karakteristik penalaan PID dengan metode <i>Ziegler-Nichols</i>	46

DAFTAR LAMPIRAN

Lampiran A	Gambar simulasi pengendali IMC untuk sistem tiga tangki terhubung	56
Lampiran B	Gambar simulasi pengendali PID untuk sistem tiga tangki terhubung	57
Lampiran C	Gambar simulasi proses nonlinear sistem tiga tangki terhubung ..	58
Lampiran D	Gambar simulasi proses nonlinear sistem tiga tangki terhubung dengan kebocoran	59

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perancangan suatu pengendali yang dapat diandalkan menjadi suatu persoalan yang cukup rumit dalam berbagai aplikasi bidang kendali. Dapat diandalkan disini berarti dapat memenuhi performa pengendalian yang diinginkan. Permasalahan sering timbul akibat proses yang akan dikendalikan merupakan proses yang rumit, bersifat nonlinear dan tidak semua karakteristik dari proses dapat diketahui atau diturunkan. Untuk mengatasi masalah ini dikembangkan berbagai skema pengendalian yang cocok untuk proses nonlinear dan juga bersifat adaptif serta *robust*, salah satunya adalah IMC [1].

Dalam mendesain suatu pengendali terlebih dahulu diawali dengan menurunkan model dari proses atau dikenal juga dengan identifikasi. Pemodelan atau identifikasi dari suatu proses merupakan salah satu langkah penting dalam pemecahan berbagai masalah ilmiah, seperti perancangan pengendali sistem. Identifikasi sistem adalah proses penurunan model dari data hasil eksperimen. Berbagai metode identifikasi dikembangkan untuk mencapai pendekatan yang optimal dari suatu sistem/proses. Perkembangan tersebut menyebabkan semakin banyaknya penggunaan metode identifikasi dalam berbagai bidang praktis, seperti biomedik, fisika elektrik, teknologi proses dan lain-lain. Hasil dari identifikasi suatu proses dikenal sebagai model [2].

Dalam perkembangannya, sistem yang dimodelkan memiliki berbagai kompleksitas tersendiri. Salah satu permasalahan yaitu sistem yang ada pada umumnya memiliki sifat yang nonlinear. Hal ini menyebabkan pemodelan dengan mengasumsikan sistem sebagai sistem yang linear atau menggunakan pendekatan linear tidak cukup mewakili sistem secara optimal pada setiap titik kerja dari proses. Oleh karena itu beberapa metode dikembangkan untuk memodelkan sifat linear dan nonlinear yang dimiliki suatu sistem. Salah satu model yang dapat mewakili kedua sifat tersebut adalah model Hammerstein [3].

Model Hammerstein merupakan salah satu model nonlinear khusus yang terdiri dari blok nonlinear statis dan diikuti oleh sistem linear. Bagian nonlinear dari proses Hammerstein dapat diwakili oleh beberapa metode seperti fuzzy *Takagi-Sugeno*, sistem *neural network*, atau *neurofuzzy*. Sedangkan bagian linearnya dapat menggunakan pemodelan linear *least square* seperti *Auto-Regresive eXogenous* (ARX), maupun pendekatan *polynomial* dan ekspansi *fourrier* [3].

Pada skripsi ini, dibahas mengenai perancangan skema pengendalian berbasis model dari suatu proses nonlinear. Model proses yang digunakan adalah model nonlinear Hammerstein. Pada model tersebut digunakan estimator *neural network* untuk merepresentasikan fungsi nonlinearnya, dan model ARX untuk mewakili bagian linearnya. Dengan menggunakan metode yang sama, diturunkan juga inverse model dari sistem tersebut, untuk menyusun pengendali dari proses. Implementasi dari skema kendali yang digunakan adalah dengan menggunakan metode *Internal Model Control* (IMC). Performa dari pengendali sangat ditentukan dari model dan invers model yang digunakan. Oleh karena itu, ditambahkan juga *Radial Basis Function Network* (RBFN), untuk meningkatkan performa dari sistem, terutama ketika model yang dihasilkan kurang mewakili proses tersebut.

1.2 Tujuan Pembahasan

Tujuan pembuatan skripsi ini adalah:

1. Melakukan proses identifikasi dari pasangan data masukan-keluaran hasil simulasi tiga tangki terhubung, untuk memperoleh model Hammerstein dari sistem tersebut.
2. Merancang pengendali Internal Model Control (IMC) berbasiskan model nonlinear Hammerstein dan invers modelnya.
3. Menganalisa dan membandingkan kinerja dari pengendali IMC yang diperoleh dengan beberapa skema pengendali lainnya.
4. Mensimulasikan kinerja dari pengendali yang diperoleh terhadap terjadinya gangguan pada sistem.

1.3 Pembatasan Masalah

Masalah yang akan dibahas pada skripsi ini meliputi identifikasi sistem dengan menggunakan metode Hammerstein pada sistem tiga tangki terhubung. Model tersebut terdiri dari neural network untuk mewakili fungsi nonlinearnya dan model ARX untuk bagian linearnya. Berdasarkan model tersebut, dirancang sebuah skema kendali Internal Model Control (IMC) untuk mengendalikan sebuah sistem tiga tangki terhubung. Selain IMC ditambahkan pula sebuah RBFN untuk meningkatkan performanya, menghilangkan *offset* dan ketahanannya terhadap *disturbance*. Sistem tiga tangki yang dibahas disini masih berupa model matematis, yang mungkin akan memiliki sedikit perbedaan dengan sistem dalam prakteknya.

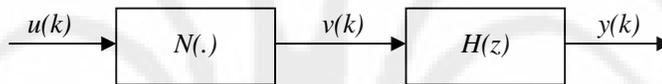
1.4 Sistematika Penulisan

Penulisan skripsi ini terdiri dari 5 (lima) bagian, bagian pertama adalah pendahuluan yang berisikan pendahuluan, terdiri dari latar belakang, tujuan pembahasan, pembatasan masalah, dan sistematika penulisan. Bagian kedua merupakan dasar teori, berisi penjelasan mengenai model proses Hammerstein, neural network, inverse model, RBFN, dan skema Internal Model Control (IMC). Bagian ketiga adalah penurunan model matematis dari sistem nonlinear tiga tangki terhubung, penurunan model nonlinear Hammerstein dari sistem tersebut, validasi dan simulasi dari model yang diperoleh, perancangan pengendali IMC. Bagian keempat berisikan hasil simulasi dari sistem tiga tangki terhubung yang dikendalikan oleh IMC yang telah diperoleh, analisa dan perbandingannya jika dibandingkan dengan pengendali yang sejenis, serta performanya ketika terjadi gangguan berupa kebocoran. Bagian kelima berupa kesimpulan yang diambil berdasarkan hasil pembahasan, simulasi dan analisa yang dilakukan pada bab-bab sebelumnya.

BAB 2 DASAR TEORI

2.1 Model Proses Hammerstein

Salah satu model yang dapat merepresentasikan baik bagian linear dan nonlinear dari suatu proses adalah model Hammerstein. Model Hammerstein merupakan model nonlinear *block-oriented* yang terdiri dari struktur *cascade* dari fungsi nonlinear statis $N(\cdot)$ yang diikuti dengan blok linear dinamis $H(z)$. Model dengan struktur ini diketahui dapat mewakili pendekatan dari berbagai proses industri seperti proses distilasi, dinamika friksi, ketinggian air atau tekanan udara dan lain-lain [5].



Gambar 2.1 Struktur model Hammerstein

Dalam persamaan matematis, model Hammerstein dapat dituliskan sebagai berikut

$$v(k) = N(u(k)) \quad (2.1)$$

$$y(k) = \sum_{i=1}^{n_a} a_i y(k-i-n_d) + \sum_{j=1}^{n_b} b_j v(k-j-n_d) \quad (2.2)$$

dimana $u(k)$, $v(k)$ dan $y(k)$ menyatakan masukan, variable antara dan keluaran proses pada waktu cuplik ke- k , a_i dan b_j merupakan parameter model n_a , n_b merupakan orde dari model linear, dan n_d merupakan jumlah delay yang digunakan pada model linear. $N(\cdot)$ merupakan fungsi nonlinear yang berkaitan dengan kenonlinearan input dan $H(z)$ fungsi linear dinamis dari proses. Untuk sistem dengan input dan output *multiple channels*, fungsi $N(\cdot)$ berbeda untuk tiap channelnya. Dimensi dari variabel antara $v(k)$ sama dengan dimensi dari input $u(k)$.

Tujuan dari pemodelan ini adalah untuk mencapai model Hammerstein yang optimal untuk *cost function* E , sehingga

$$E(N(\cdot), a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}) = \frac{1}{p} \sum_{k=1}^p (\hat{y}(k) - y(k))^2 < \epsilon \quad (2.3)$$

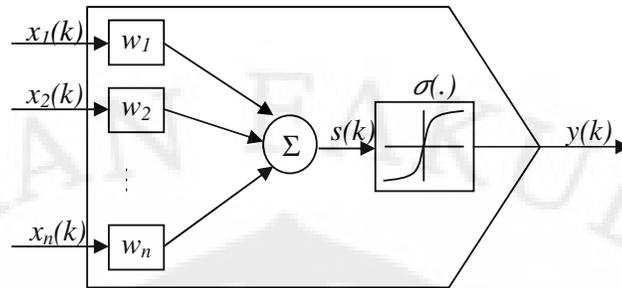
E dapat diterima untuk nilai ϵ yang kecil. Dimana y adalah keluaran dari proses Hammerstein, \hat{y} adalah estimasi keluaran model, dan p mewakili jumlah pasangan data masukan keluaran.

Pada tulisan ini, proses Hammerstein bagian nonlinear dari proses yaitu $N(\cdot)$ diidentifikasi dengan menggunakan estimator *neural network*, dan bagian linear $H(z)$ diwakili oleh struktur *Auto Regressive eXogenous* (ARX) orde-2 dengan parameter yang diestimasi dengan metode *least square*. Baik bagian nonlinear $N(\cdot)$ dan bagian linear $H(z)$ pada model ini diasumsikan memiliki parameter-parameter yang tidak berubah atau model-model tersebut bersifat statis. Penurunan model ini dilakukan secara iterative sampai mencapai nilai *cost function* E yang lebih kecil dari ϵ yang diinginkan atau mencapai jumlah iterasi maksimum.

Dalam pengembangannya, model Hammerstein ini bila ditambahkan lagi blok fungsi nonlinear setelah blok linearnya (*nonlinearity output*) menjadi model nonlinear Hammerstein-Wiener. Pada model Hammerstein, digunakan sebuah unit gain sebagai blok *nonlinearity output*-nya.

2.1.1. Neural Network

Pada model nonlinear Hammerstein ini, neural network digunakan sebagai estimator untuk bagian nonlinear dari proses. Neural network merupakan jaringan yang menyerupai jaringan sel saraf dan akan mengestimasi variabel antara $v(k)$ dari masukan model $u(k)$. Neural network terdiri dari gabungan beberapa *neuron* dengan parameternya masing-masing yang berbeda dan berfungsi sebagai estimator nonlinear.



Gambar 2.2 Struktur neuron tunggal

Gambar 2.2 menunjukkan struktur sebuah neuron tunggal dimana $x_1(k)$, $x_2(k)$, ..., $x_n(k)$ merupakan masukan dari neuron, w_1 , w_2 , ..., w_n adalah koefisien bobot neuron, $s(k)$ adalah *weighed sum* dari neuron, $\sigma(\cdot)$ dan $y(k)$ masing-masing adalah fungsi aktivasi dan keluaran dari neuron. Input yang berupa vector dikalikan dengan bobot masing-masing dan kemudian dijumlah menghasilkan *weighed sum* $s(k)$. Keluaran dari neuron $y(k)$ merupakan hasil pemetaan fungsi aktivasi $\sigma(\cdot)$ terhadap *weighed sum* $s(k)$ tersebut. Atau secara matematis dapat dituliskan hubungan antara masukan dan keluaran dari sebuah neuron tunggal sebagai berikut:

$$s(k) = \sum_{i=1}^n w_i x_i \quad (2.4)$$

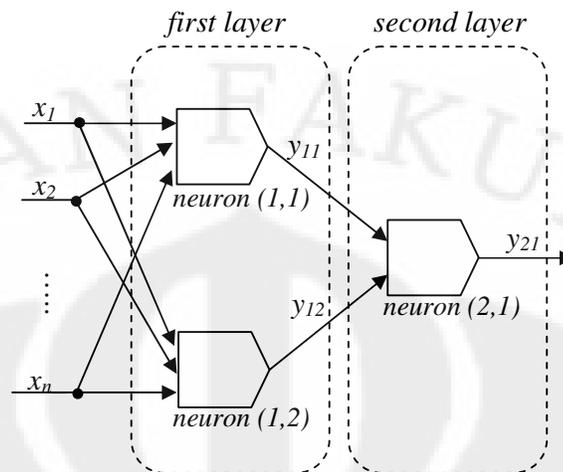
$$y(k) = \sigma(s(k)) \quad (2.5)$$

Persamaan (2.4) dan (2.5) bila dituliskan dalam bentuk vektor menjadi

$$s(k) = \mathbf{w}^T \mathbf{x} \quad (2.6)$$

$$y(k) = \sigma(s(k)) = \sigma(\mathbf{w}^T \mathbf{x}) \quad (2.7)$$

Fungsi aktivasi $\sigma(\cdot)$ pada neuron dapat berbeda-beda sesuai dengan jenis output yang diinginkan pada neural network tersebut. Sedangkan model Hammerstein pada tulisan ini menggunakan fungsi nonlinear sigmoid yang akan dibahas pada bagian selanjutnya. Demikian juga dengan algoritma pencarian koefisien bobot untuk tiap neuron.



Gambar 2.3 Topologi neural network

Sebuah neural network terdiri dari sekumpulan neuron, yang terdistribusi dalam beberapa lapisan atau *layer*. Gambar 2.3 menunjukkan topologi sebuah neural network yang terdiri dari dua buah layer dan tiga buah neuron.

2.1.2. Fungsi Aktivasi Nonlinear Sigmoid

Fungsi aktivasi adalah fungsi yang memetakan kondisi neural dari $x \in \mathcal{R}^n$ ke dalam ruang keluaran neural. Ruang keluaran neural yang dimaksudkan dapat berupa daerah $[0,1]$ untuk jenis sinyal *unipolar*, dan $[-1,1]$ untuk jenis sinyal *bipolar*. Untuk model neural network yang kontinu terhadap waktu, fungsi nonlinear sigmoid yang dapat digunakan adalah fungsi yang kontinu dan dapat diturunkan. Selain itu fungsi aktivasi juga harus memiliki karakter sebagai berikut:

- $\sigma(x)$ memiliki nilai batas atas dan nilai batas bawah.
- $\sigma(x)$ menuju nilai batasnya ketika $x \rightarrow \pm\infty$ (memiliki asimtot).
- $\sigma(x)$ sama dengan nilai tengahnya ketika $x \rightarrow 0$.
- $\sigma'(x) > 0$ dan $\sigma'(x) \rightarrow 0$ ketika $x \rightarrow \pm\infty$ (naik secara monoton).
- $\sigma'(x)$ memiliki nilai maksimum.

Fungsi aktivasi yang digunakan dalam identifikasi ini adalah fungsi *log-sigmoidal (logsig)*

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.8)$$

Beberapa fungsi aktivasi lain yang dapat digunakan adalah *signum*, *tanh*, *tansig* atau *pureli*. Fungsi aktivasi *logsig* dipilih karena memenuhi karakteristik yang tersebut diatas, dapat memfasilitasi karakteristik nonlinear dari sistem dan bekerja pada daerah output $[0,1]$.

2.1.3. Backpropagation pada Neural Network

Backpropagation merupakan salah satu algoritma belajar pada neural network. Tujuannya adalah mencapai nilai parameter bobot yang sesuai, sehingga model neural network yang dihasilkan dapat meniru proses yang diinginkan. Hal ini dapat dicapai dengan meminimalkan *Mean Square Error* (MSE) dari neural network $e^2(k)$. Kesalahan yang terjadi didefinisikan sebagai perbedaan keluaran neural network $y(k)$ dengan nilai yang diinginkan $d(k)$

$$\begin{aligned} e(k) &\triangleq d(k) - y(k) \\ &= d(k) - \sigma(s(k)) \\ &= d(k) - \sigma(\mathbf{w}^T \mathbf{x}(k)) \end{aligned} \quad (2.9)$$

Estimasi gradien sesaat pada masukan vektor input dapat dituliskan sebagai

$$\nabla_{\mathbf{w}_a} (e^2(k)) = \frac{\partial e^2(k)}{\partial \mathbf{w}_a(k)} = 2e(k) \frac{\partial e(k)}{\partial \mathbf{w}_a(k)} \quad (2.10)$$

Dengan menurunkan persamaan (2.9) diperoleh

$$\frac{\partial e(k)}{\partial \mathbf{w}_a(k)} = -\frac{\partial \sigma(s(k))}{\partial \mathbf{w}_a(k)} = -\sigma'(s(k)) \frac{\partial s(k)}{\partial \mathbf{w}_a(k)} \quad (2.11)$$

dan karena

$$s(k) = \mathbf{x}_a^T(\mathbf{k}) \mathbf{w}_a(\mathbf{k}) \quad (2.12)$$

maka

$$\frac{\partial s(k)}{\partial \mathbf{w}_a(k)} = \mathbf{x}_a(k) \quad (2.13)$$

Kemudian dengan mensubstitusikan persamaan (2.13) ke persamaan (2.11) maka diperoleh

$$\frac{\partial e(k)}{\partial \mathbf{w}_a(k)} = -\sigma'(s(k))\mathbf{x}_a(k) \quad (2.14)$$

Dengan hasil ini maka persamaan (2.10) menjadi

$$\nabla_{\mathbf{w}_a} (e^2(k)) = -2e(k)\sigma'(s(k))\mathbf{x}_a(k) \quad (2.15)$$

Sehingga vektor koefisien bobot dari neural network pada waktu cuplik selanjutnya $w_a(k+1)$ diperoleh

$$\begin{aligned} \mathbf{w}_a(k+1) &= \mathbf{w}_a(k) - \nabla_{\mathbf{w}_a} (e^2(k)) \\ &= \mathbf{w}_a(k) + \alpha e(k)\sigma'(s(k)) \end{aligned} \quad (2.16)$$

dimana α merupakan *learning rate* dari model yang nilainya berkisar pada [0,2] yang nilainya dapat ditala secara manual. Dengan menggunakan persamaan tersebut, maka untuk sejumlah *learning data* dapat diperoleh nilai koefisien bobot \mathbf{w}_a dengan MSE minimum.

2.1.4. Sigmoid Network

Sigmoid network digunakan sebagai estimator nonlinear pada model Hammerstein ini. Sigmoid network merupakan neural network khusus dengan fungsi aktivasi sigmoid. Fungsi alih dari sigmoid network ini adalah sebagai berikut

$$\begin{aligned} y &= N(x) \\ N(x) &= (x-r)\mathbf{P}\mathbf{I} + a_1\sigma((x-r)\mathbf{Q}\mathbf{b}_1 - c_1) + a_2\sigma((x-r)\mathbf{Q}\mathbf{b}_2 - c_2) + \dots \\ &\quad \dots + a_n\sigma((x-r)\mathbf{Q}\mathbf{b}_n - c_n) + d \end{aligned} \quad (2.17)$$

dimana σ adalah fungsi aktivasi sigmoid yang dijelaskan pada persamaan (2.8). \mathbf{P} dan \mathbf{Q} adalah matriks proyeksi berdimensi $(m \times p)$ dan $(m \times q)$ yang ditentukan dari analisis data estimasi. Bila data estimasi yang digunakan bebas linear maka $p=m$, jika tidak maka $p < m$. Sedangkan jumlah kolom \mathbf{Q} , yaitu q bergantung pada

banyaknya dimensi masukan \mathbf{x} yang digunakan dalam fungsi sigmoidnya, karena sistem yang dimodelkan berupa sistem *single input single output* (SISO), maka $q=1$. Dan untuk model Hammerstein $m=q=1$ sehingga \mathbf{P} dan \mathbf{Q} adalah scalar pada model ini. r adalah vektor ($1 \times m$) yang mewakili nilai tengah dari vektor *regressor* yang dihitung dari data estimasi. Kemudian \mathbf{I} merupakan vektor koefisien linear yang berdimensi ($p \times 1$). n merupakan banyaknya neuron pada hidden layer dari sigmoid network yang juga menggambarkan ukuran kenonlinearan dari sistem. b_k, c_k, a_k masing-masing adalah faktor dilasi, faktor translasi dan koefisien keluaran dari sistem, sedangkan d merupakan koefisien *offset* berupa skalar.

Semua parameter-parameter dari sigmoid network ini diperoleh dari data estimasi, dengan terlebih dahulu menentukan parameter kenonlinearannya (n). Keluaran dari fungsi ini berupa variabel antara $v(k)$, menjadi input untuk blok linear ARX dari model Hammerstein.

2.1.5. Model Linear ARX

ARX adalah kepanjangan dari *Auto-Regressive with eXogenous input*, yang dapat diartikan sebuah model *auto regressive* dengan masukan *exogenous*, atau masukan yang berasal dari luar model yang dibentuk. Model ARX juga termasuk suatu sistem *linear time invariance* (LTI), dengan struktur sebagai berikut

$$y(k) = \frac{B(z)}{A(z)}u(k) \quad (2.18)$$

dimana z merupakan operator delay, $u(k)$ dan $y(k)$ masing-masing adalah masukan dan keluaran dari model pada waktu cuplik ke- k . Model dengan struktur ini dapat mengestimasi keluaran proses secara linear sebagai fungsi dari keluaran dan masukan sebelumnya

$$\begin{aligned} y(k) = & -a_1y(k-1) - a_2y(k-2) - \dots - a_{n_a}y(k-n_a) + \dots \\ & \dots + b_1u(k-1) + b_2u(k-2) + \dots + b_{n_b}u(k-n_b) \end{aligned} \quad (2.19)$$

dengan a_1, a_2, \dots, a_n dan b_1, b_2, \dots, b_n merupakan parameter model ARX, sedangkan n_a dan n_b merupakan orde model.

Persamaan (2.19) dapat ditulis dalam bentuk matriks dan vektor sebagai berikut

$$\underbrace{\begin{bmatrix} y(n_a+1) \\ y(n_a+2) \\ \vdots \\ y(N) \end{bmatrix}}_{\hat{\mathbf{y}}} = \underbrace{\begin{bmatrix} -y(n_a) & -y(n_a-1) & \cdots & -y(1) & u(n_b) & u(n_b-1) & \cdots & u(1) \\ -y(n_a+1) & -y(n_a) & \cdots & -y(2) & u(n_b-1) & u(n_b) & \cdots & u(2) \\ \vdots & \vdots \\ -y(N-1) & -y(N-2) & \cdots & -y(N-n_a) & u(N-1) & u(N-2) & \cdots & u(N-n_b) \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n_a} \\ b_1 \\ b_2 \\ \vdots \\ b_{n_b} \end{bmatrix}}_{\theta} \quad (2.20)$$

$$\hat{\mathbf{y}} = \Phi \theta$$

dengan N adalah banyaknya *training* data.

Untuk memperoleh vektor nilai parameter θ , digunakan sejumlah training data untuk membuat nilai MSE dari model ini minimum. MSE yang minimum dapat diperoleh dengan terlebih dahulu membuat persamaan [6]

$$\begin{aligned} e^2 &= \sum_{i=1}^N (y_i - \hat{y}_i)^2 \\ &= (\mathbf{y} - \Phi \theta)^T (\mathbf{y} - \Phi \theta) \\ &= \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \Phi \theta - (\Phi \theta)^T \mathbf{y} + (\Phi \theta)^T \Phi \theta \\ &= \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \Phi \theta - \theta^T \Phi^T \mathbf{y} + \theta^T \Phi^T \Phi \theta \end{aligned} \quad (2.21)$$

Nilai parameter θ yang dapat membuat persamaan (2.21) minimum merupakan nilai parameter θ yang membuat turunan dari persamaan tersebut bernilai nol. Turunan dari persamaan (2.21)

$$\begin{aligned} \frac{de^2}{d\theta} &= \frac{d}{d\theta} (\mathbf{y}^T \mathbf{y} - \mathbf{y}^T \Phi \theta - \theta^T \Phi^T \mathbf{y} + \theta^T \Phi^T \Phi \theta) \\ 0 &= 0 - (\mathbf{y}^T \Phi)^T - \Phi^T \mathbf{y} + \theta [(\Phi^T \Phi)^T + (\Phi^T \Phi)] \\ \theta &= (\Phi^T \Phi)^{-1} \Phi^T \mathbf{y} \end{aligned} \quad (2.22)$$

Nilai parameter θ tersebut kemudian digunakan untuk mengestimasi bagian linear dinamis dari proses, yaitu

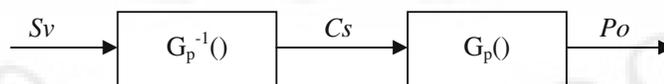
$$\hat{y} = \Phi\theta \quad (2.23)$$

Masukan dari blok linear ini adalah keluaran dari blok nonlinear sigmoid network pada pembahasan sebelumnya. Dan keluaran dari blok linear ini merupakan hasil estimasi keseluruhan dari model Hammerstein ini.

2.2 Inverse Model Hammerstein

Inverse atau kebalikan dari model Hammerstein secara umum merupakan model Hammerstein yang dilatih untuk menghasilkan proses yang berkebalikan dengan model referensinya. Hal ini dapat dilakukan dengan menggunakan data masukan sebagai keluaran dan data keluaran sebagai masukan. Dengan demikian akan dihasilkan model yang memiliki karakteristik yang berkebalikan dengan model proses referensinya. Data masukan keluaran yang digunakan dapat diperoleh dari plant secara langsung, namun bila hal tersebut tidak dapat dilakukan data dapat diperoleh dari model referensi yang telah diperoleh terlebih dahulu. Dengan cara ini, kinerja inverse model yang dihasilkan akan sangat bergantung dari model referensinya.

Inverse dari model dapat langsung dijadikan sebagai pengendali dari proses, karena memiliki karakteristik yang berkebalikan dengan proses, sehingga masukan berupa setpoint (*setpoint value/Sv*) diubah menjadi sinyal kendali (*control signal/Cs*) yang akan menghasilkan setpoint tersebut kemudian diteruskan ke proses. Dengan demikian, keluaran dari proses akan menghasilkan nilai output (*process output/Po*) yang sesuai dengan setpoint yang diberikan. Skema kendali seperti ini dikenal sebagai skema *open loop control*.



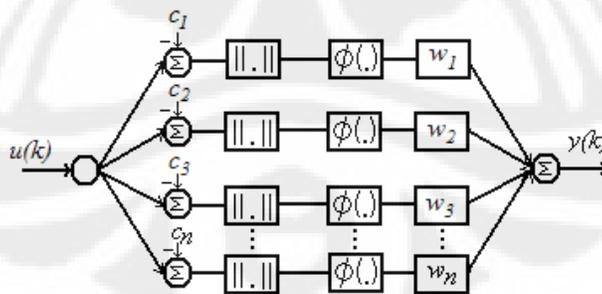
Gambar 2.4 Skema open loop control

Dengan menggunakan skema seperti ini, kinerja dari pengendali bergantung pada inverse dari model proses yang dihasilkan. Bila proses memiliki karakteristik, letak pole yang saling menghilangkan, maka kinerja pengendali akan menjadi baik. Namun proses yang ada pada umumnya bersifat nonlinear dan sulit untuk mengestimasi keberadaan pole dari proses tersebut, sehingga terjadinya kesalahan dan *offset* tidak dapat dihindarkan. Selain itu skema kendali seperti ini juga sangat rentan terhadap gangguan pada proses serta perubahan parameter atau karakteristik dari proses. Dan dalam proses yang terjadi pada umumnya, kedua hal ini sangat mungkin terjadi.

Untuk mengatasi kendala yang timbul, maka disertakan umpan balik (*feedback*) pada sistem. Sehingga pengendali yang digunakan dapat beradaptasi terhadap adanya gangguan atau perubahan pada sistem dan juga dapat menghilangkan *offset* yang terjadi akibat ketidaksempurnaan model yang digunakan. Skema pengendali yang digunakan pada tulisan ini adalah Internal Model Control/IMC, yang akan dijelaskan pada pembahasan selanjutnya.

2.3 Radial Basis Function Network (RBFN)

Radial Basis Function Network atau RBFN termasuk salah satu bentuk alternatif dari *feedforward neural network*. RBFN adalah fungsi nonlinear multidimensional yang memetakan dengan bergantung pada jarak dari input ke titik tengah dari sistem atau disebut juga dengan *cluster*.



Gambar 2.5 Radial Basis Function Network

Sebuah RBFN dengan masukan u berdimensi n , $u \in \mathfrak{R}^n$ dan keluaran tunggal digambarkan pada gambar 6.1. Secara matematis, RBFN tersebut dapat dituliskan sebagai berikut [7]:

$$y = f(u) = \sum_{i=1}^n w_i \phi(\|u - c_i\|) \quad (2.24)$$

Pada persamaan tersebut $\phi(\|u - c_i\|)$ merupakan fungsi radial basis dari u , yang nilainya bergantung pada jarak input u terhadap titik tengah cluster c_i . Fungsi $\phi(\cdot)$ merupakan fungsi yang nonlinear, $\|\cdot\|$ menandakan norma Euclidean. Fungsi radial basis yang digunakan pada RBFN di tulisan ini adalah fungsi radial Gaussian sebagai berikut

$$\phi(u) = \exp\left(-\frac{1}{2}\left[\frac{u - c}{\sigma}\right]^2\right) \quad (2.25)$$

σ dan c masing-masing merupakan lebar dan titik tengah dari tiap cluster. Untuk memperoleh parameter-parameter dari RBFN ini maka terlebih dahulu digunakan pasangan data masukan keluaran untuk melatih neural network tersebut. Sedangkan algoritma yang digunakan untuk memperoleh nilai-nilai titik tengah dari cluster sebagai berikut

- a. Data masukan adalah $u(k)$ ($k=1,2,\dots,P$), titik tengah dari cluster pertama c_1 didefinisikan sebagai $u(1)$ ($c_1 = u(1)$). Dengan demikian jumlah data masukan pada cluster pertama N_1 dan jumlah cluster N adalah masing-masing $N_1 = 1$ dan $N = 1$.
- b. Untuk setiap *training* data $u(k)$, kemiripan dari training data ke- k dibandingkan dengan semua titik tengah cluster c_l ($l = 1,2,\dots,N$) dan cluster yang memiliki titik tengah terdekat dengan $u(k)$ adalah cluster dengan karakteristik berikut:

$$S_L = \max_{1 \leq l \leq N} \sqrt{\exp\left(-\|u(k) - c_l\|^2\right)} \quad (2.26)$$

- c. Tentukan apakah cluster (aturan fuzzy) yang baru perlu dibentuk atau tidak dengan menggunakan criteria sebagai berikut:

- Jika $S_L < \beta$, dimana merupakan batasan yang telah ditetapkan, *training* data ke- k tidak masuk ke cluster yang ada dan cluster baru dibentuk dengan titik tengah $c_{N+1} = u(k)$. Set $N = N+1$ dan $N_N = 1$.
- Jika $S_L \geq \beta$, maka *training* data ke- k masuk ke cluster ke- L dan cluster ke- L akan disesuaikan sebagai berikut

$$c_L = c_L + \lambda(u(k) - c_L) \quad (2.27)$$

$$\lambda = \frac{\lambda_o}{N_L + 1}, \lambda_o \in [0,1] \quad (2.28)$$

$$N_L = N_L + 1 \quad (2.29)$$

- Kemudian $k = k + 1$ dan ulangi langkah (b) dan (c) sampai semua *training* data masuk ke cluster yang bersangkutan. Setelah itu, jumlah cluster dan aturan fuzzy menjadi tetap yaitu N , dan lebar dari *fuzzy set* dihitung dengan:

$$\sigma_L = \min_{j=1,2,\dots,N, j \neq l} |c_l - c_j| / \rho \quad (2.30)$$

dimana ρ adalah parameter *overlap* ($1 \leq \rho \leq 2$).

Selanjutnya adalah pengestimasi parameter bobot untuk RBFN ini. Hal ini dilakukan dengan cara mencari solusi nilai parameter bobot \mathbf{w} yang meminimalkan error yang terjadi. Misalkan keluaran dari RBFN adalah:

$$\Phi \mathbf{w} = \mathbf{y} \quad (2.31)$$

dimana Φ merupakan fungsi jarak Gaussian yang digunakan dalam bentuk matriks. Maka error yang terjadi adalah

$$\mathbf{E} = \|\mathbf{y} - \mathbf{w}\|^2 \quad (2.32)$$

Persamaan tersebut memiliki nilai error yang minimum ketika parameter bobotnya bernilai \mathbf{w}^* dan

$$\mathbf{w}^* = \Phi^+ \mathbf{y} \quad (2.33)$$

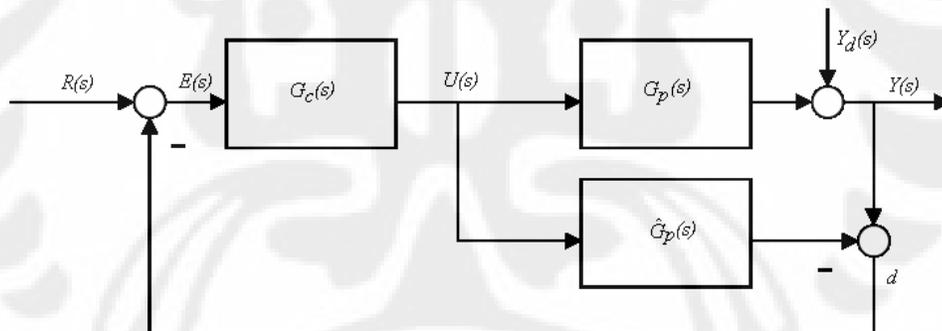
dimana Φ^+ mewakili matriks inverse general dari Φ yaitu

$$\Phi^+ = [\Phi^T \Phi]^{-1} \Phi^T \quad (2.34)$$

Dengan demikian dengan menggunakan pasangan data input output dapat diperoleh nilai w . Parameter dan data mengenai cluster yang diperoleh kemudian digunakan untuk simulasi dan validasi dari model. Pada pengaplikasiannya, parameter bobot w juga dapat diperbarui tiap waktu cupliknya dengan menggunakan persamaan (2.16).

2.4 Internal Model Control (IMC)

Skema IMC yang umum terdiri dari tiga blok, seperti terlihat pada gambar 2.6 yaitu blok inverse model Hammerstein, blok model Hammerstein dan blok proses. Pada skema pengendalian ini, selisih antara keluaran dan model diumpanbalikan untuk mengurangi nilai acuan yang hasilnya menjadi masukan dari pengendali.



Gambar 2.6 Internal Model Control

Penempatan model Hammerstein secara paralel dengan proses dimaksudkan untuk mengurangi efek dari gangguan pada keluaran dari proses dan kesalahan yang terjadi pada pemodelan. Jika model dapat mewakili secara sempurna karakteristik dari proses atau memiliki parameter-parameter yang sama dengan proses, maka tidak ada umpanbalik yang terjadi. Sehingga dapat dikatakan sistem adalah lingkaran terbuka. Selanjutnya bila inverse dari model dapat

merepresentasikan kebalikan dari proses yang ada pada sistem, maka kutub pada kedua proses ini akan saling menghilangkan dan keluaran dari sistem tersebut akan sama dengan nilai acuan yang diberikan.

Berikut adalah penjelasan tentang skema IMC dalam domain s . Jika $G_p(s) = \tilde{G}_p(s)$ maka $d(s)$ adalah gangguan dari luar yang tidak diketahui asalnya. Maka $d(s)$ dapat dianggap sebagai informasi yang hilang dari model, dan dapat digunakan untuk memperbaiki proses pengendalian yaitu dengan mengurangi $d(s)$ terhadap $R(s)$ sehingga hasil dari sinyal pengendali adalah sebagai berikut:

$$\begin{aligned} U(s) &= [R(s) - d(s)]G_c(s) \\ &= \{R(s) - [G_p(s) - \tilde{G}_p(s)]U(s) - d(s)\}G_c(s) \end{aligned} \quad (2.35)$$

dari persamaan (2.35) dapat diubah menjadi

$$U(s) = \frac{[R(s) - d(s)]G_c(s)}{1 + [G_p(s) - \tilde{G}_p(s)]G_c(s)} \quad (2.36)$$

Karena

$$Y(s) = G_p(s)U(s) + d(s) \quad (2.37)$$

maka fungsi dari *loop* tertutupnya adalah

$$Y(s) = \frac{G_c(s)G_p(s)R(s) + [1 - G_c(s)\tilde{G}_p(s)]d(s)}{1 + [G_p(s) - \tilde{G}_p(s)]G_c(s)} \quad (2.38)$$

Analisa ini merupakan analisa dalam domain s , pada aplikasinya model yang digunakan dapat berupa juga model nonlinear seperti model Hammerstein.

Dalam aplikasinya, IMC juga dapat diterapkan pada domain diskrit. Hal ini dikarenakan penekanan pada skema kendali ini adalah bagaimana cara menghasilkan model (\tilde{G}_p) dan invers model (G_c) yang dapat mewakili proses dan invers dari proses nonlinear yang terjadi pada sistem (G_p). Model maupun invers

model tersebut selain dapat berupa model di domain s , juga dapat berupa model pada domain diskrit ataupun model nonlinear.

Hal ini dikarenakan bila model dan invers model diasumsikan bekerja secara sempurna mewakili proses dan invers proses sistem, maka keluaran model dan keluaran sistem akan menjadi sama sehingga tidak ada umpan balik yang terjadi pada sistem atau dengan kata lain sistem menjadi bersifat lingkaran terbuka. Kemudian bila invers dari model juga mewakili dengan sempurna kebalikan proses yang terjadi pada sistem, maka dengan diberikannya nilai acuan berupa ketinggian fluida yang diinginkan, maka keluaran dari invers model adalah sinyal kendali berupa tegangan pompa yang membuat sistem menghasilkan ketinggian fluida sesuai dengan nilai acuan tersebut. Dengan demikian keluaran sistem akan menjadi sama dengan nilai acuan yang diberikan dan dapat disimpulkan bahwa model dan invers model yang digunakan dapat bervariasi selama masih mampu mewakili proses dan invers proses yang terjadi pada sistem.

BAB 3
PERANCANGAN PENGENDALI INTERNAL MODEL CONTROL
UNTUK SISTEM TIGA TANGKI TERHUBUNG

3.1 Penurunan Model Nonlinear Sistem Tiga Tangki Terhubung

Suatu sistem yang berhubungan dengan fluida bergerak, atau disebut sistem hidrodinamika, dapat dijelaskan dengan menggunakan sistem Bernoulli. Salah satu contohnya adalah pada sistem tiga tangki terhubung. Penurunan model ini dilakukan untuk memperoleh pasangan data masukan keluaran yang akan digunakan untuk identifikasi dan simulasi.

Sistem tiga tangki yang digunakan diasumsikan memiliki keadaan fluida yang tunak (*steady*), tidak memiliki kepekatan (*nonviscous*) dan tanpa tekanan atau tak termampatkan (*incompressible*). Dengan demikian persamaan Bernoulli dapat dituliskan sebagai berikut:

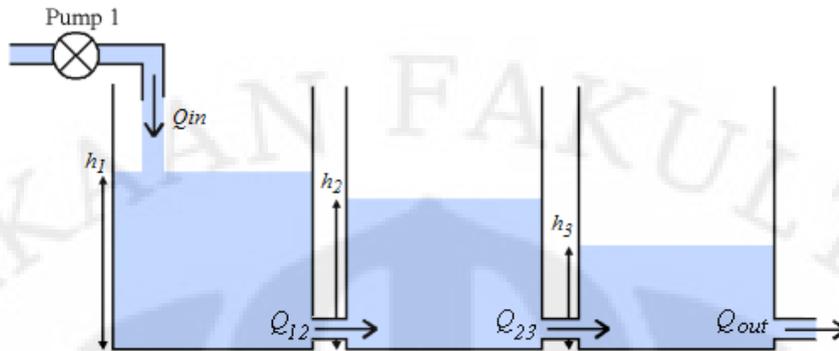
$$p_1 + \frac{1}{2}\rho v_1^2 + \rho g h_1 = p_2 + \frac{1}{2}\rho v_2^2 + \rho g h_2 \quad (3.1)$$

dengan

- p_1, p_2 : tekanan fluida (dyne/cm²)
- ρ : massa jenis fluida (gram/cm³)
- v_1, v_2 : kecepatan fluida (cm/s)
- h_1, h_2 : ketinggian fluida (cm)

dapat digunakan untuk memodelkan sistem tiga tangki terhubung, yang merupakan suatu sistem hidrodinamika, karena berhubungan dengan pergerakan keadaan fluida.

Sistem tiga tangki terhubung seperti terlihat pada gambar 3.1 merupakan konfigurasi dari pemodelan yang akan digunakan untuk menghasilkan data masukan-keluaran dari suatu proses nonlinear.



Gambar 3.1 Sistem tiga tangki terhubung

Persamaan kecepatan aliran fluida yang keluar dari setiap tangki dapat ditentukan dengan menggunakan persamaan Torricelli:

$$v = \sqrt{2gh} \quad (3.2)$$

yang menyatakan hubungan antara kecepatan aliran dengan ketinggian fluida pada tangki, yaitu kecepatan fluida yang berbanding lurus dengan akar kuadrat dari ketinggian fluida.

Untuk memodelkan perilaku dari sistem tiga tangki terhubung sehingga perubahan pada ketinggian fluida dapat diprediksi berdasarkan kondisi kecepatan alirannya, dengan mengabaikan efek dari temperatur terhadap fluida, sehingga kesetimbangan energi tidak diperhitungkan, maka kesetimbangan massa untuk setiap tangki memiliki persamaan umum sebagai berikut:

$$\frac{d(\rho Ah)}{dt} = \rho Q_{in} - \rho Q_{out}, \quad (3.3)$$

dengan demikian kondisi keadaan setiap tangki pada sistem tiga tangki terhubung dapat dinyatakan dengan persamaan:

$$\frac{A_1 dh_1}{dt} = Q_{in} - Q_{12}, \quad (3.4)$$

$$\frac{A_2 dh_2}{dt} = Q_{12} - Q_{23}, \quad (3.5)$$

$$\frac{A_3 dh_3}{dt} = Q_{23} - Q_{out}, \quad (3.6)$$

yang merupakan persamaan keadaan ketinggian fluida untuk tangki pertama (3.4), tangki kedua (3.5) dan tangki ketiga (3.6). Setiap variabel yang mempengaruhi ketinggian fluida, yaitu Q_{1out} , Q_{2out} , dan Q_{3out} ditentukan oleh ketinggian permukaan fluida disetiap tangki:

$$Q_{12} = a_1 \operatorname{sgn}(h_1 - h_2) \sqrt{|h_1 - h_2|}, \quad (3.7)$$

$$Q_{23} = a_2 \operatorname{sgn}(h_2 - h_3) \sqrt{|h_2 - h_3|}, \quad (3.8)$$

$$Q_{out} = a_3 \sqrt{h_3}, \quad (3.9)$$

dengan

A_1, A_2, A_3 : luas penampang tangki (cm^2)

h_1, h_2, h_3 : ketinggian fluida dari titik referensi (cm)

a_1, a_2, a_3 : luas penampang pipa katup (cm^2)

Q_{in} : debit fluida yang masuk ke tangki pertama (cm^3/s)

Q_{1out} : debit fluida yang keluar dari tangki pertama (cm^3/s)

Q_{2out} : debit fluida yang keluar dari tangki kedua (cm^3/s)

Q_{3out} : debit fluida yang keluar dari tangki ketiga (cm^3/s)

$d(\rho Ah)$: perubahan volume fluida pada tangki (cm^3)

dt : perubahan waktu (s)

Sedangkan $\operatorname{sgn}(\cdot)$ merupakan fungsi signum yang memetakan masukannya ke nilai 1 dan -1.

$$\operatorname{sgn}(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0. \end{cases} \quad (3.10)$$

Karena tidak adanya efek panas yang dapat menyebabkan perubahan volume pada fluida dan pada tangki itu sendiri, maka massa jenis dari fluida akan selalu konstan dan demikian pula dengan luas alas penampang. Dengan demikian,

persamaan keadaan setiap tangki, berdasarkan persamaan differensial (3.4), (3.5), (3.6), dinyatakan dalam persamaan:

$$\frac{dh_1}{dt} = \frac{Q_{in} - a_1 \operatorname{sgn}(h_1 - h_2) \sqrt{|h_1 - h_2|}}{A_1}, \quad (3.11)$$

$$\frac{dh_2}{dt} = \frac{a_1 \operatorname{sgn}(h_1 - h_2) \sqrt{|h_1 - h_2|} - a_2 \operatorname{sgn}(h_2 - h_3) \sqrt{|h_2 - h_3|}}{A_2}, \quad (3.12)$$

$$\frac{dh_3}{dt} = \frac{a_2 \operatorname{sgn}(h_2 - h_3) \sqrt{|h_2 - h_3|} - a_3 \sqrt{h_3}}{A_3}. \quad (3.13)$$

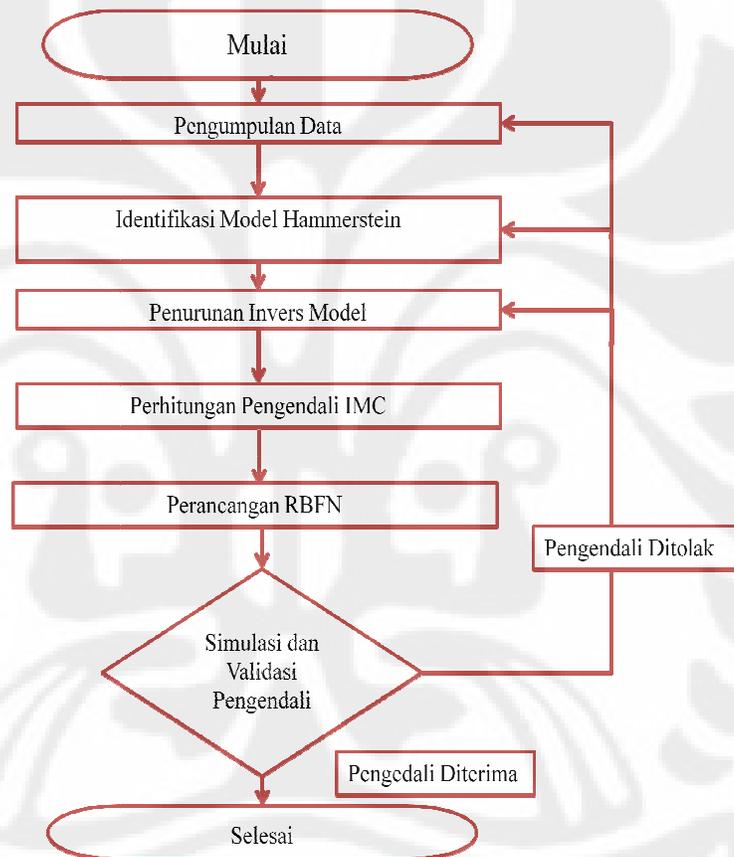
Persamaan diatas merupakan persamaan nonlinear yang menyatakan keadaan ketinggian fluida untuk setiap tangki pada sistem tiga tangki terhubung yang dipakai untuk menggenerasi data masukan-keluaran melalui simulasi dengan menggunakan *software Simulink 7.1* dari *MATLAB 7.6*. Masukan yang dimaksud pada percobaan ini adalah sinyal tegangan pada pompa yang mengatur aliran fluida ke tangki pertama dan keluarannya adalah ketinggian air pada tangki ketiga. Dengan demikian sistem yang akan dimodelkan adalah *single input single output* (SISO) dari sistem tiga tangki terhubung [8].

Berikut adalah spesifikasi dari sistem tiga tangki terhubung yang digunakan dalam simulasi

Tabel 3.1 Spesifikasi sistem tiga tangki terhubung

Parameter	Keterangan	Nilai
h_1	Tinggi tangki 1	50 cm
h_2	Tinggi tangki 2	40 cm
h_3	Tinggi tangki 3	15 cm
A_1	Luas penampang tangki 1	100 cm ²
A_2	Luas penampang tangki 2	45 cm ²
A_3	Luas penampang tangki 3	30 cm ²

a_1	Luas Penampang pipa katup 1	0.5 cm^2
a_2	Luas Penampang pipa katup 2	0.4 cm^2
a_3	Luas Penampang pipa katup 3	0.35 cm^2
G	Konstanta gravitasi	980 cm/s^2
$K_{vq} (Q_{in} = K_{vq} \times V_{pump})$	<i>Volt to flow factor</i>	$6 \text{ cm}^3/\text{Vs}$
V_{pump}	Interval tegangan pompa	$2 \leq V_{pump} \leq 10 \text{ V}$



Gambar 3.2 Prosedur perancangan pengendali IMC

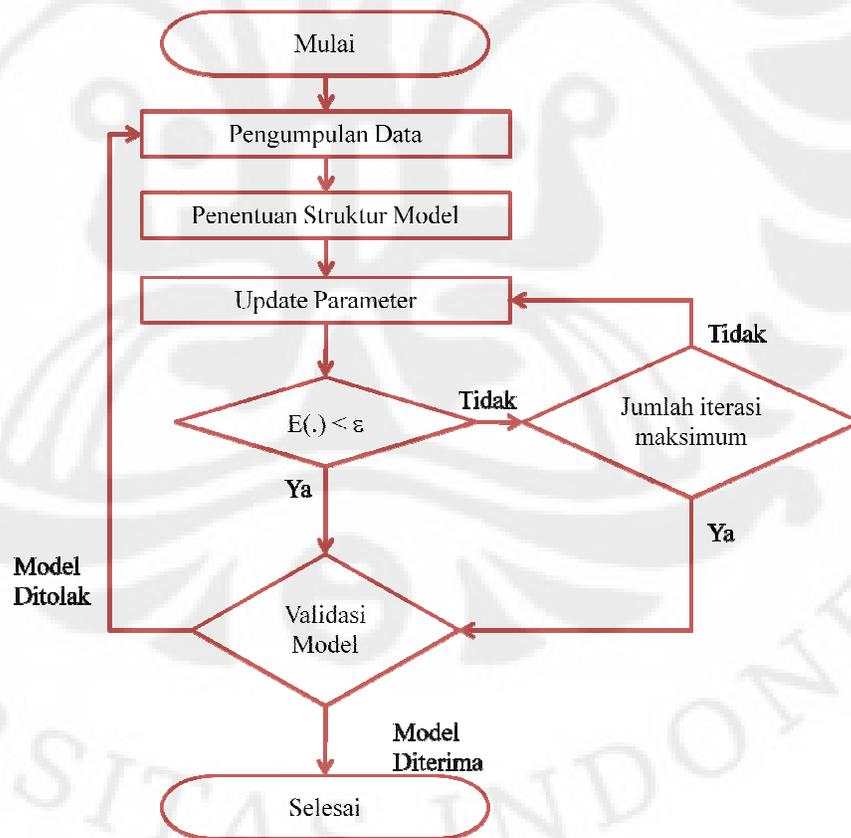
Setelah diperoleh model matematis dari sistem tiga tangki terhubung ini, maka pasangan data masukan keluaran dapat diperoleh dan perancangan pengendali dapat dilakukan dengan menggunakan prosedur seperti yang ditunjukkan pada gambar 3.2.

3.2 Identifikasi Model Hammerstein

Pada bagian ini dan selanjutnya akan dibahas mengenai proses identifikasi dari sistem tiga tangki terhubung dengan menggunakan pasangan data masukan keluaran sampai dihasilkan model nonlinear Hammerstein. Bagian nonlinear statis dari sistem akan diwakili oleh blok neural network sigmoidal, sedangkan bagian linear dinamisnya diwakili oleh blok *autoregressive exogenous* (ARX).

Sistem tiga tangki terhubung yang digunakan merupakan suatu sistem yang ideal, dimana faktor-faktor yang mempengaruhi aliran fluida dianggap konstan untuk setiap waktu, seperti hubungan tegangan pompa ke debit fluida yang dipompakan ke dalam tangki pertama (*volt to flow factor*), yaitu sebesar $6 \text{ cm}^3/\text{Vs}$. selain itu sistem tiga tangki terhubung yang dibahas juga hanya masih merupakan model matematis, yang mungkin akan mengalami cukup banyak perubahan pada implementasinya.

Gambar 3.3 menunjukkan algoritma yang digunakan dalam proses identifikasi sistem tiga tangki terhubung.



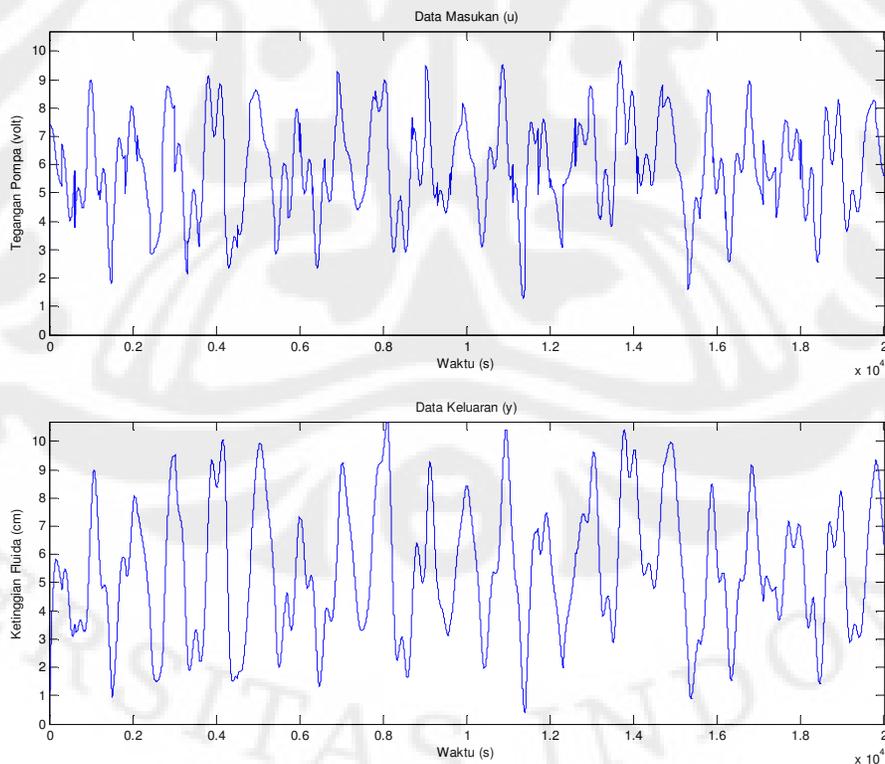
Gambar 3.3 Algoritma identifikasi model Hammerstein

dimana E merupakan cost function dari sistem, yang menentukan selesainya proses iterasi bila nilainya cukup kecil ($< \epsilon$). Bila hal tersebut tidak dapat tercapai, maka iterasi akan terus dilakukan sampai jumlah iterasi maksimum tercapai. Setelah seluruh parameter model diperoleh maka dilakukan validasi untuk menguji kinerja dari model yang diperoleh.

3.2.1 Pengumpulan Data Sistem Tiga Tangki Terhubung

Langkah pertama dalam identifikasi model Hammerstein adalah dengan mengambil pasangan data masukan keluaran pada sistem tiga tangki yang digunakan. Data masukan keluaran tersebut diperoleh dengan memberikan nilai masukan multisinusoidal dengan dengan *range* interval tegangan pompa yaitu [2,10] volt, dan kemudian mencatat keluaran yaitu tinggi air pada tangki ke tiga. Data diambil selama 20000 detik, dengan demikian diperoleh data masukan keluaran yang cukup untuk memodelkan sistem tiga tangki tersebut.

Gambar 3.4 menunjukkan grafik dari data masukan dan keluaran yang digunakan untuk mengidentifikasi sistem tiga tangki terhubung

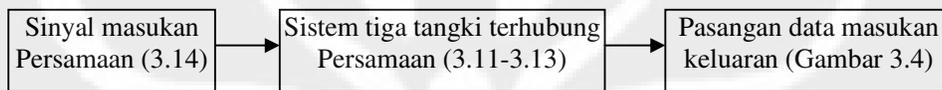


Gambar 3.4 Data masukan dan keluaran untuk identifikasi sistem

Persamaan (3.14) menunjukkan persamaan matematis dari data masukan ke sistem tiga tangki terhubung yang digunakan untuk identifikasi sistem:

$$u(t) = \left[0.5 \sin\left(\frac{0.06}{\pi}t + 90\right) \times 3 \sin\left(\frac{0.004}{\pi}t\right) \right] + 2 \sin\left(\frac{0.02}{\pi}t + 90\right) + \dots + 4.5u(t) + e(t) \quad (3.14)$$

dimana fungsi $e(t)$ merupakan nilai acak dengan interval $[0,3]$ dengan lebar sinyal untuk setiap gelombang acak sebesar 300 detik. Adapun blok diagram untuk pengambilan data dapat dilihat pada gambar 3.5.



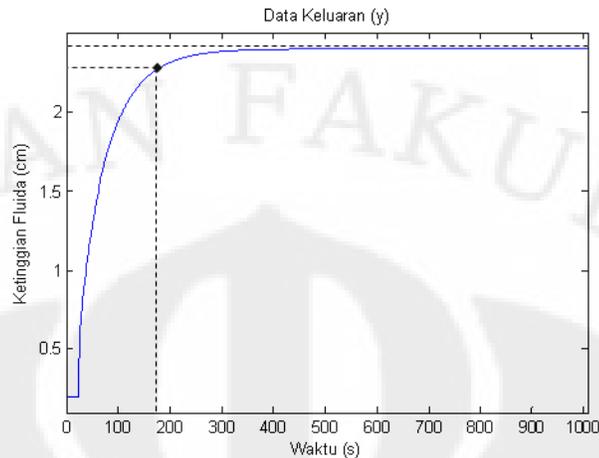
Gambar 3.5 Diagram blok pengambilan pasangan data training

3.2.2 Penentuan Waktu Pencuplikan

Penentuan waktu pencuplikan juga dapat mempengaruhi kinerja dari sistem. Interval waktu pencuplikan yang terlalu kecil sebaiknya tidak dilakukan untuk menghindari perubahan masukan proses yang terlalu besar. Selain itu, waktu pencuplikan yang terlalu besar juga dapat menyebabkan sinyal diskrit yang dihasilkan oleh pengendali tidak terekonstruksi dengan baik menjadi sinyal kontinu yang menjadi masukan proses. Untuk memperoleh rentang waktu untuk pencuplikan (h) yang baik, maka dapat digunakan persamaan [6]

$$\frac{1}{20}T_{95} \leq h \leq \frac{1}{5}T_{95} \quad (3.15)$$

dimana T_{95} adalah *settling time*, atau waktu yang dibutuhkan keluaran untuk mencapai nilai 95% dari nilai akhirnya. Untuk mendapatkan waktu pencuplikan tersebut, maka yang dapat dilakukan adalah mencari nilai tanggapan waktu dari sistem lingkaran terbuka pada model proses dengan masukan step 4 Volt. Kurva respons tanggapan dari sistem lingkaran terbuka dari sistem tiga tangki terhubung ditunjukkan pada gambar 3.6.



Gambar 3.6 Respons tanggapan sistem lingkak terbuka

dari gambar 3.6, dapat dilihat nilai tunak dari sistem pada keadaan lingkak terbuka adalah 2.4 cm. Dengan demikian 95% dari nilai tunak tersebut adalah 2.28 cm dan waktu yang dibutuhkan untuk mencapai nilai tersebut adalah 179.6 detik. berdasarkan persamaan (3.15) maka rentang waktu pencuplikan yang dapat digunakan untuk sistem ini adalah

$$\frac{1}{20} \times 179.6 \leq h \leq \frac{1}{5} \times 179.6$$

$$8.98 \leq h \leq 35.92 \quad (3.16)$$

Dari rentang waktu pencuplikan tersebut, maka untuk sistem ini digunakan waktu pencuplikan $h = 10$ detik.

3.2.3 Penurunan Parameter-parameter Model Hammerstein

Penurunan parameter-parameter model Hammerstein ini dilakukan secara iterasi sampai mencapai nilai *cost function* yang lebih kecil dari nilai batas ϵ . Kriteria *cost function* yang digunakan dijelaskan pada persamaan (2.3). Teknik ini disebut juga dengan optimasi, dan untuk menentukan arah *trajectory* dari pergerakan parameter-parameter model, digunakan metode Gauss-Newton.

Metode Gauss-Newton merupakan salah satu pendekatan untuk menentukan arah pencarian dalam metode iterasi yang digunakan. Dalam metode

ini arah *trajektori* (d_k) dari parameter dapat dicari dengan menyelesaikan persamaan

$$\mathbf{J}(\mathbf{x}_k)\mathbf{d}_k = -F(\mathbf{x}_k) \quad (3.17)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k \quad (3.18)$$

dimana $F(\cdot)$ adalah fungsi yang akan diminimumkan, dan \mathbf{J} merupakan matriks segiempat *Jacobian* dari $F(\cdot)$.

$$\mathbf{J}(\mathbf{x}_k) = \begin{bmatrix} \nabla F_1(\mathbf{x}_k)^T \\ \nabla F_2(\mathbf{x}_k)^T \\ \vdots \\ \nabla F_n(\mathbf{x}_k)^T \end{bmatrix} \quad (3.19)$$

Dengan demikian parameter-parameter dari model Hammerstein ini dapat diperoleh. Pada model ini digunakan nilai batas $\varepsilon = 0.01$ dan jumlah iterasi maksimum adalah 100 iterasi.

3.2.4 Struktur Model Hammerstein

Setelah pasangan data masukan keluaran diperoleh, maka parameter-parameter model dapat diturunkan. Untuk bagian nonlinear dari sistem digunakan model sigmoidal neural network dengan fungsi yang dijelaskan pada persamaan (2.17). Tabel 3.2 menunjukkan hasil dari penurunan parameter-parameter sigmoid neural network untuk mewakili bagian nonlinear dari sistem tiga tangki terhubung.

Tabel 3.2 Parameter-parameter dari blok nonlinear model Hammerstein

Parameter	Nilai
<i>Number of Units (n)</i>	10
<i>Regressor Mean (r)</i>	5.3043
<i>Nonlinear Subspace (Q)</i>	1
<i>Linear Subspace (P)</i>	1
<i>Linear Coeficient (L)</i>	0.1546

<i>Dilation Factor</i> (b_1, b_2, \dots, b_n)	$10^{-3} \begin{bmatrix} 0.0203 \\ 0.0023 \\ 0.0303 \\ 0.0133 \\ 0.0004 \\ 0.0057 \\ 0.0344 \\ 0.0164 \\ 1.0164 \\ 0.0175 \end{bmatrix}^T$
<i>Translation Factor</i> (c_1, c_2, \dots, c_n)	$10^{-3} \begin{bmatrix} 0.0770 \\ 0.0090 \\ 0.0648 \\ 0.0138 \\ -0.0003 \\ -0.0045 \\ -0.0431 \\ -0.0242 \\ -2.4184 \\ -0.0565 \end{bmatrix}^T$
<i>Output Coefficient</i> (a_1, a_2, \dots, a_n)	$\begin{bmatrix} 0.0636 \\ -0.2924 \\ -0.0360 \\ -0.0052 \\ -0.3091 \\ 0.0273 \\ 0.0026 \\ 0.0291 \\ 0.0586 \\ 0.0337 \end{bmatrix}$
<i>Output Offset</i> (d)	0.7538

Sedangkan untuk blok linearnya, digunakan struktur ARX orde-2 ($n_a = 2$, $n_b = 2$) diperoleh nilai parameter-parameter sebagai berikut

$$\boldsymbol{\theta} = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -0.7619 \\ -0.2239 \\ -0.8299 \\ 1 \end{bmatrix} \quad (3.20)$$

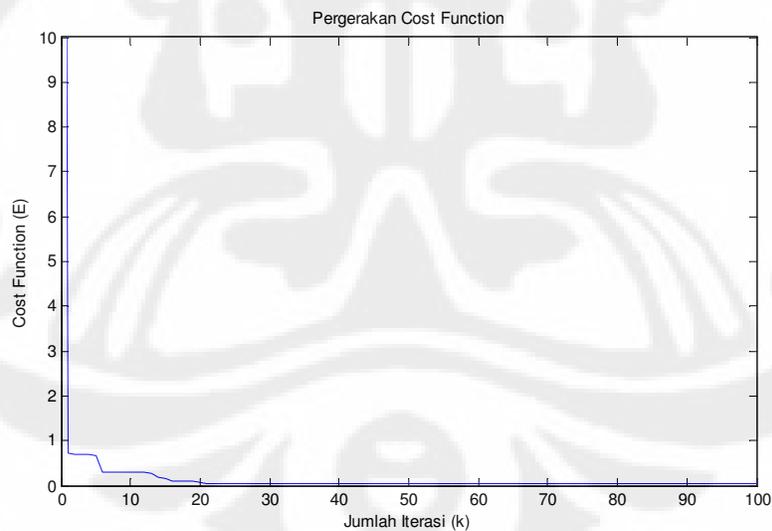
atau dalam persamaan domain z , dapat dituliskan

$$y(z) = \frac{-0.8299z^{-1} + z^{-2}}{1 - 0.7619z^{-1} - 0.2239z^{-2}} u(z) \quad (3.21)$$

Dengan z adalah operator *delay*. Maka hubungan keluaran pada waktu cuplik ke- k dengan masukan dan keluaran pada waktu cuplik sebelumnya adalah

$$y(k) = 0.7619y(k-1) + 0.2239y(k-2) - 0.8299u(k-1) + \dots \\ \dots + u(k-2) \quad (3.22)$$

Nilai parameter-parameter ini diperoleh setelah 100 iterasi dengan nilai *cost function* sebesar 0.056.



Gambar 3.7 Pergerakan *cost function* terhadap jumlah iterasi

Gambar 3.7 menunjukkan pergerakan *cost function* terhadap jumlah iterasi. Terlihat bahwa nilai *cost function* konvergen menuju nilai minimum seiring dengan bertambahnya jumlah iterasi sesuai dengan penjelasan pada bagian

sebelumnya. Karena nilai batas untuk *cost function* $\varepsilon = 0.01$ tidak dapat tercapai, maka proses iterasi berhenti pada jumlah iterasi maksimum yaitu 100 iterasi.

3.3 Validasi Model Hammerstein

Setelah mendapatkan model Hammerstein, maka perlu dilakukan validasi untuk mengetahui performa dari model. Proses validasi dilakukan dengan cara melakukan perbandingan antara sinyal keluaran proses dengan sinyal keluaran yang diestimasi oleh model Hammerstein tersebut. Kemudian dari dua data tersebut dapat dihitung parameter yang menentukan bagus atau tidaknya model yang diperoleh. Parameter tersebut antara lain *Root Mean Square Error* (RMSE), yaitu

$$RMSE = \sqrt{\frac{\sum (y - y_m)^2}{N}} \quad (3.23)$$

dan *Variable Accounted For* (VAF)

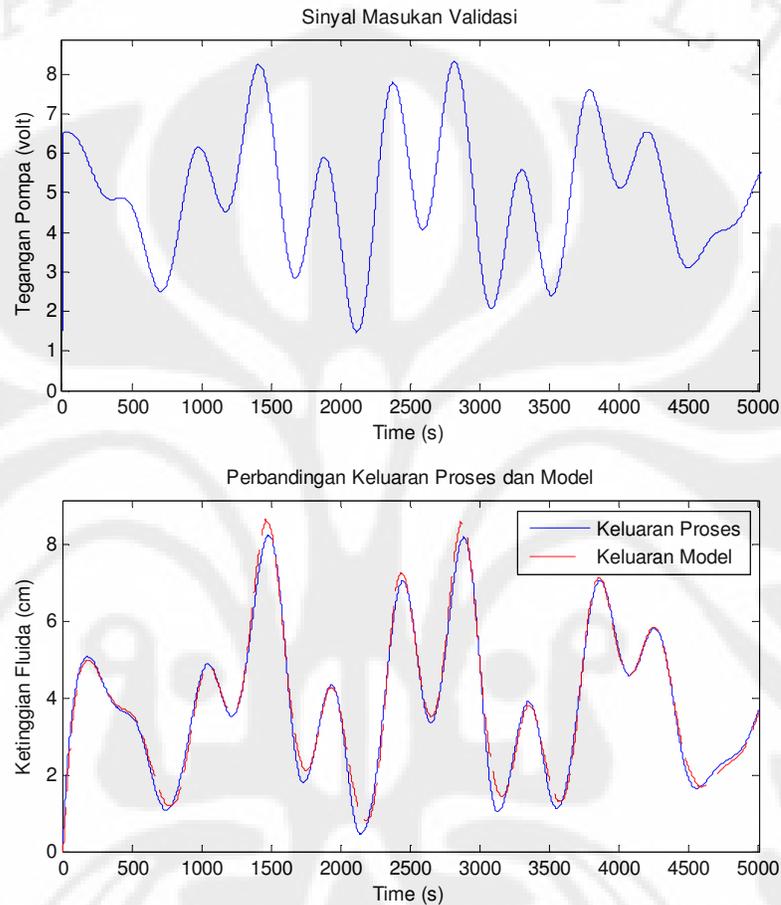
$$VAF = 100\% \times \left[1 - \frac{\text{var}(y - y_m)}{\text{var}(y)} \right] \quad (3.24)$$

dimana y dan y_m masing-masing merupakan keluaran dari proses dan estimasi keluaran dari model Hammerstein. Sedangkan N adalah jumlah pasangan data yang digunakan dalam validasi ini. Model yang baik menunjukkan nilai RMSE yang cukup kecil, sedangkan untuk VAF mendekati 100%.

Untuk memvalidasi model perlu digunakan data masukan keluaran yang berbeda dengan data identifikasi. Pada proses validasi ini, digunakan dua jenis data, yaitu data masukan keluaran dengan sinyal masukan multifrekuensi, dan data masukan keluaran dengan sinyal masukan fungsi step.

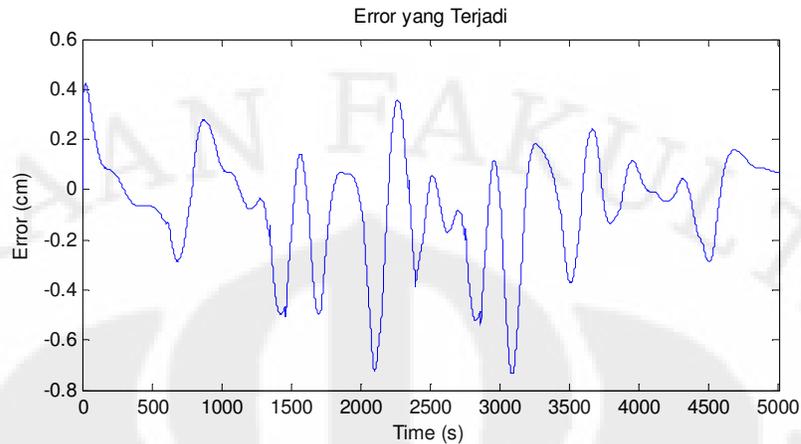
$$u(t) = \left[2 \sin\left(\frac{0.042}{\pi}t + 90\right) \times 1.2 \sin\left(\frac{0.002}{\pi}t\right) \right] + 1.5 \sin\left(\frac{0.015}{\pi}t + 90\right) + \dots + 5u(t) \quad (3.25)$$

Gambar 3.8 menunjukkan grafik sinyal masukan serta perbandingan antara keluaran proses dan keluaran model untuk validasi model Hammerstein ini. Data diambil selama 5000 detik.



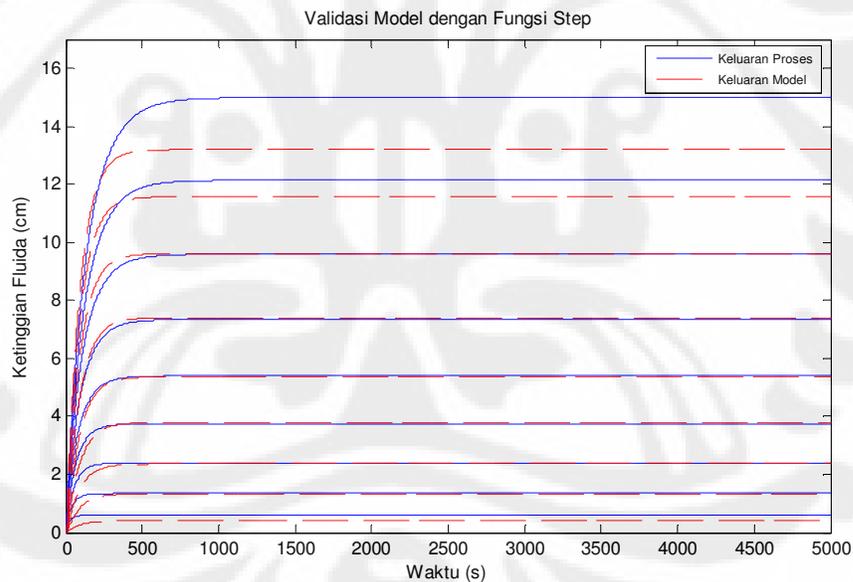
Gambar 3.8 Validasi model dengan masukan sinyal multifrekuensi

Dari gambar 3.8 dapat terlihat secara visual keluaran dari model dapat tidak berbeda jauh dengan keluaran dari proses. Dengan demikian dapat dikatakan bahwa parameter-parameter yang diperoleh sebelumnya menghasilkan model Hammerstein yang cukup mewakili proses yang sebenarnya. Pernyataan ini juga didukung dengan dihasilkannya RMSE dan VAF masing-masing sebesar 0.2258 dan 98.69%. Nilai RMSE yang cukup kecil dan VAF yang mendekati 100% tersebut menandakan bahwa kinerja model cukup baik.



Gambar 3.9 Kesalahan pada validasi sinyal multifrekuensi

Gambar 3.9 menunjukkan kesalahan yang terjadi pada validasi sinyal multifrekuensi yang merupakan selisih antara keluaran proses yang sebenarnya dengan keluaran yang diestimasi model. Kesalahan yang terjadi berkisar pada nilai antara -0.738 sampai 0.4206.



Gambar 3.10 Validasi model dengan sinyal masukan fungsi step

Selain menggunakan sinyal multifrekuensi sebagai input, proses validasi juga dilakukan dengan sinyal masukan step, karena dengan sinyal masukan step dapat terlihat kesalahan pada kondisi tunak dari sistem. Gambar 3.10 menunjukkan hasil validasi berupa step respons dari proses dan model. Sinyal masukan fungsi step yang digunakan adalah $2u(t)$, $3u(t)$, ..., $10u(t)$, yang mencakup seluruh daerah

Universitas Indonesia

kerja masukan dari sistem. Dari gambar terlihat bahwa model bekerja cukup baik di daerah sekitar 1-10 cm, yaitu masukan 3 V sampai 8 V. Hal ini disebabkan data yang digunakan untuk mengestimasi parameter memiliki rata-rata pada daerah tersebut, sehingga pada daerah itu model bekerja baik dalam mewakili proses. Pada masukan 10 V keluaran proses tepat mengalami saturasi pada ketinggian 15 cm.

Tabel 3.3 Nilai RMSE dan VAF pada proses validasi model

Sinyal Masukan	RMSE	VAF
Multifrekuensi	0.2258	98.69 %
$2u(t)$	0.1937	40.80 %
$3u(t)$	0.0701	46.41 %
$4u(t)$	0.0926	80.87 %
$5u(t)$	0.0852	92.59 %
$6u(t)$	0.0692	98.59 %
$7u(t)$	0.0761	99.52 %
$8u(t)$	0.1224	98.68 %
$9u(t)$	0.5503	95.96 %
$10u(t)$	1.7302	91.03 %

Tabel 3.3 menunjukkan nilai RMSE dan VAF untuk seluruh sinyal validasi. Dari tabel tersebut terlihat model dapat bekerja dengan baik, terutama pada daerah 3 V sampai 8 V. Selanjutnya, model ini akan dimanfaatkan untuk membuat invers dari model yang akan digunakan di pengendali IMC.

3.4 Perancangan Invers Model Hammerstein

Perancangan invers dari model Hammerstein dilakukan dengan algoritma yang serupa dengan perancangan model Hammerstein. Yaitu dengan cara mengumpulkan data masukan keluaran yang membuat model dapat meniru kebalikan dari proses yang terjadi pada model referensi. Kemudian mengestimasi parameter-parameter model dengan cara iterasi hingga *cost function*-nya bernilai kurang dari nilai batas yang diinginkan, yaitu sebesar $\varepsilon = 0.01$ dan dengan jumlah

iterasi maksimum sebanyak 100 iterasi. Selain itu struktur dari invers model ini juga sama dengan model referensinya.

Untuk invers model Hammerstein ini, diperoleh parameter-parameter untuk blok nonlinearnya seperti pada tabel 3.4

Tabel 3.4 Parameter-parameter dari blok nonlinear dari invers model Hammerstein

Parameter	Nilai
<i>Number of Units (n)</i>	10
<i>Regressor Mean (r)</i>	5.2187
<i>Nonlinear Subspace (Q)</i>	1
<i>Linear Subspace (P)</i>	1
<i>Linear Coefficient (L)</i>	45.3762
<i>Dilation Factor (b_1, b_2, \dots, b_n)</i>	$\begin{bmatrix} 16.3114 \\ 1.2241 \\ 23.7994 \\ 0.9569 \\ -501.5306 \\ -0.8825 \\ 6.7748 \\ 7.6471 \\ 0.3148 \\ 3.7749 \end{bmatrix}^T$
<i>Translation Factor (c_1, c_2, \dots, c_n)</i>	$\begin{bmatrix} 76.3105 \\ 6.7733 \\ 57.7531 \\ 5.3144 \\ -460.6139 \\ -1.1064 \\ -4.0774 \\ -14.0823 \\ -1.1637 \\ -19.6823 \end{bmatrix}^T$

<i>Output Coefficient</i> (a_1, a_2, \dots, a_n)	$\begin{bmatrix} 4.9072 \\ 242.4202 \\ -0.2251 \\ -311.2499 \\ -0.1498 \\ 42.2154 \\ -0.2026 \\ -0.2516 \\ -464.4532 \\ -3.1308 \end{bmatrix}$
<i>Output Offset</i> (d)	272.5893

Sedangkan untuk parameter blok linearnya adalah

$$\theta = \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \end{bmatrix} = \begin{bmatrix} -0.8667 \\ 0.1019 \\ 1 \\ -0.9875 \end{bmatrix} \quad (3.26)$$

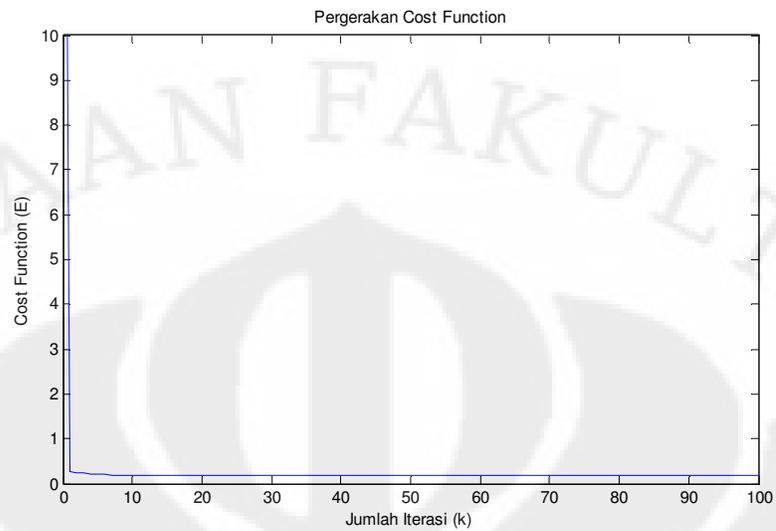
atau dalam persamaan domain z , dapat dituliskan

$$y(z) = \frac{z^{-1} - 0.9875z^{-2}}{1 - 0.8667z^{-1} + 0.1019z^{-2}} u(z) \quad (3.27)$$

maka hubungan keluaran pada waktu cuplik ke- k dengan masukan dan keluaran pada waktu cuplik sebelumnya adalah

$$\begin{aligned} y(k) = & 0.8667y(k-1) - 0.1019y(k-2) + u(k-1) + \dots \\ & \dots - 0.9875u(k-2) \end{aligned} \quad (3.28)$$

Nilai-nilai parameter ini diperoleh setelah iterasi ke-100 dengan *cost function* sebesar -0.1728. Gambar 3.11 menunjukkan pergerakan dari *cost function* sampai iterasi ke-100.



Gambar 3.11 Pergerakan *cost function* terhadap jumlah iterasi pada penurunan invers model

Selanjutnya, invers model Hammerstein ini akan digunakan dalam perancangan pengendali dengan skema IMC.

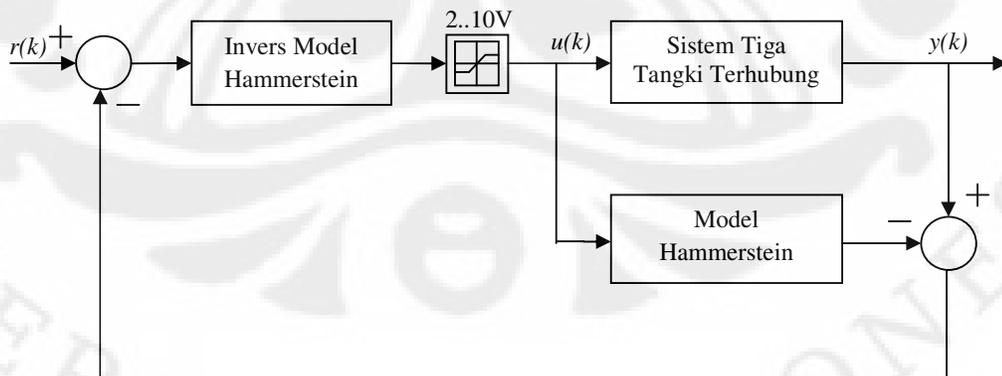
BAB 4

UJI COBA SIMULASI DAN ANALISA

4.1 Simulasi Pengendali IMC untuk Sistem Tiga Tangki Terhubung

Pada sub bab ini dibahas mengenai simulasi pengendalian pada sistem tiga tangki terhubung dengan menggunakan pengendali Internal Model Control (IMC). Pada IMC terdapat blok model, invers model dan plant, ketiga blok tersebut telah diperoleh pada pembahasan sebelumnya. Model yang digunakan pada skema ini adalah model nonlinear Hammerstein. Model tersebut terdiri dari blok nonlinear, yang diwakili oleh neural network dalam persamaan (2.17), dan blok linear yang diwakili oleh estimator dengan struktur ARX dalam persamaan (2.23). Nilai-nilai parameter yang dibutuhkan untuk kedua persamaan tersebut tercantum dalam tabel 3.2 dan persamaan (3.20).

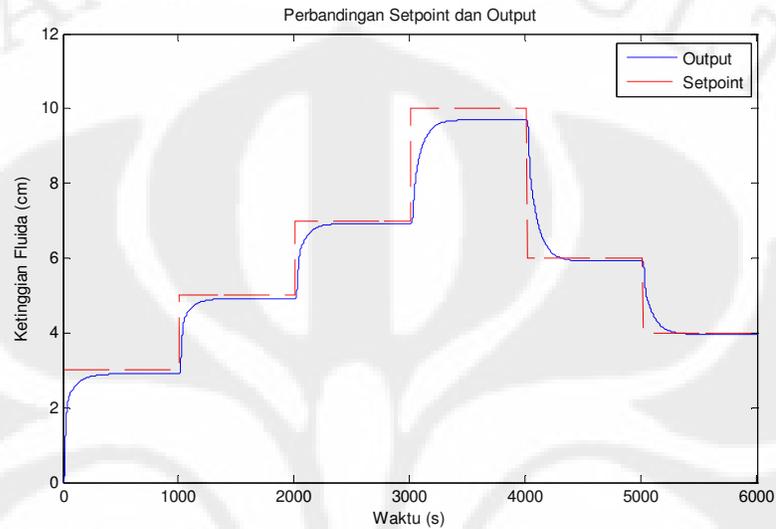
Proses yang dikendalikan pada simulasi ini adalah ketinggian tangki ketiga dalam sistem tiga tangki terhubung, yang dalam bentuk model matematis dinyatakan dalam persamaan (3.12). Selain itu spesifikasi dari sistem tiga tangki terhubung yang digunakan dapat dilihat pada tabel 3.1. Simulasi pengendalian dengan skema IMC ini dilakukan dengan menggunakan perangkat lunak SIMULINK 7.1 yang merupakan bagian dari MATLAB 7.6. Gambar 4.1 menunjukkan diagram blok yang digunakan pada simulasi.



Gambar 4.1 Diagram blok yang digunakan pada simulasi pengendali IMC

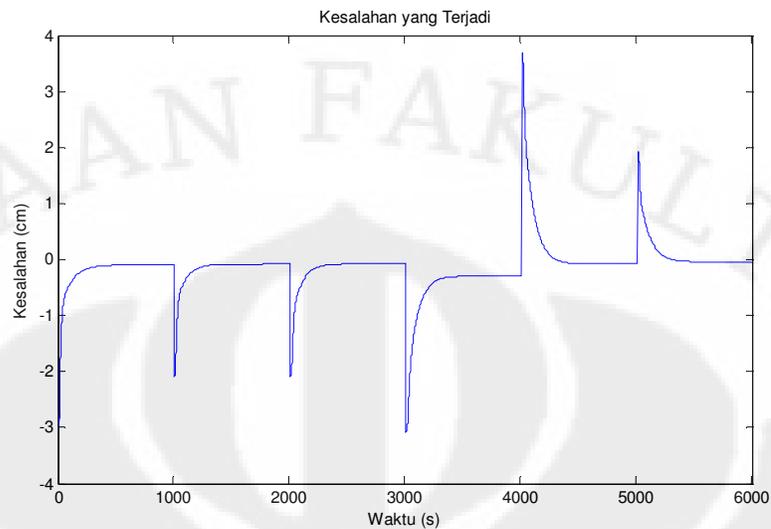
Pada diagram blok tersebut juga ditambahkan blok saturasi yang bertujuan membatasi sinyal kendali yang masuk ke sistem tiga tangki terhubung. Hal ini

dikarenakan spesifikasi masukan yang dapat diterima sistem tersebut adalah pada interval [2,10] volt. Hasil dari simulasi diagram blok tersebut dapat dilihat pada gambar 4.2.



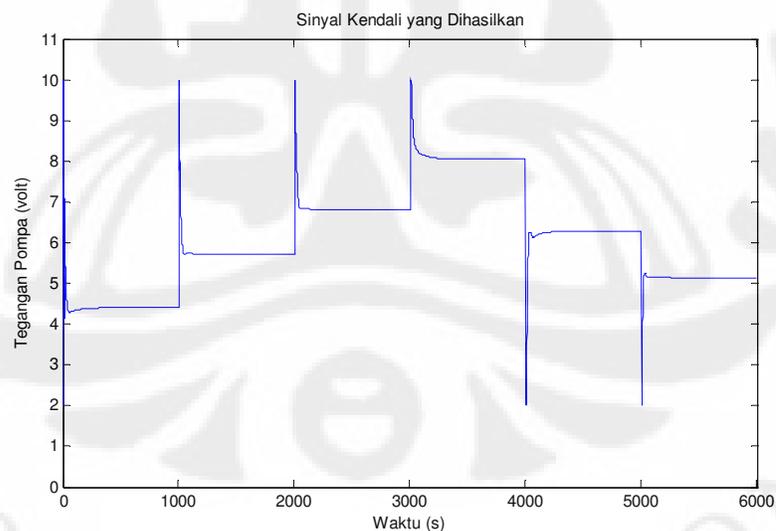
Gambar 4.2 Hasil simulasi pengendalian IMC pada sistem tiga tangki terhubung

Dari gambar 4.2 terlihat bahwa keluaran dari sistem dapat mengikuti nilai acuan yang diberikan. Sehingga dapat dikatakan pengendali yang digunakan bekerja dengan baik. Namun masih terdapat beberapa kekurangan yang terjadi pada pengendali ini, yaitu masih adanya kesalahan galat tunak pada sistem, yang terjadi terutama pada pinggiran daerah kerja dari sistem yang besarnya mencapai 0.3 cm pada nilai acuan 10 cm. Hal ini disebabkan invers model yang digunakan tidak sepenuhnya berlawanan dengan proses yang ada, dengan kata lain kutub-kutub pada kedua proses tersebut tidak saling menghilangkan sehingga terjadi perbedaan dari respon galat tunak dengan nilai acuan yang diberikan. Untuk lebih jelasnya, dapat dilihat pada gambar 4.3, yang menunjukkan bahwa kesalahan yang terjadi pada simulasi pengendalian IMC ini tidak mencapai angka nol, namun masih berkisar di angka yang cukup kecil pada keadaan tunaknya.



Gambar 4.3 Kesalahan yang terjadi pada simulasi sistem dengan pengendali IMC

Adapun penyebab invers model tidak sepenuhnya mewakili kebalikan dari proses pada sistem adalah karena persebaran data yang digunakan pada identifikasi sistem lebih banyak pada bagian tengah dari daerah kerja sistem yaitu dengan nilai *mean* 5.69 V. Dengan demikian, invers model bekerja lebih baik pada daerah 3 cm sampai 9 cm, dibandingkan daerah lainnya.



Gambar 4.4 Sinyal kendali IMC yang menjadi masukan sistem tiga tangki terhubung

Gambar 4.4 menunjukkan grafik sinyal kendali $u(t)$ yang dihasilkan oleh pengendali IMC dan menjadi masukan untuk sistem tiga tangki terhubung. Dari

grafik tersebut dapat dilihat bahwa sinyal kendali yang masuk berkisar pada daerah [2,10] volt, sesuai dengan daerah kerja masukan sistem. Pengendali juga menunjukkan sinyal kendali dengan respons yang cukup cepat terhadap perubahan nilai acuan yang diberikan. Selain itu, sinyal kendali juga konvergen menuju ke suatu nilai dalam waktu yang singkat. Hal ini menandakan pengendali IMC yang diperoleh sudah bekerja cukup baik, terlepas dari kesalahan galat tunak yang dihasilkan.

4.2 Perancangan *Radial Basis Function Network* (RBFN)

Untuk mencapai hasil pengendalian yang baik dengan menggunakan skema IMC, maka model dan invers model yang dihasilkan harus dapat meniru proses dan invers proses dari sistem yang ingin dikendalikan pada semua daerah kerjanya. Hal ini cukup sulit untuk dilakukan terutama ketika sistem yang dikendalikan adalah sistem nonlinear. Untuk mengatasi permasalahan ini, dapat digunakan suatu blok tambahan berupa blok *Radial Basis Function Network* (RBFN) secara *cascade* dengan pengendali IMC dan sistem.

Perancangan RBFN ini sesuai dengan pembahasan pada sub bab 2.3 dengan menggunakan parameter-parameter β , λ , dan ρ masing-masing sebesar 0.6, 0.1, dan 1.5. Maka diperoleh RBFN dengan titik tengah dan lebar *cluster* yang tercantum pada tabel 4.1.

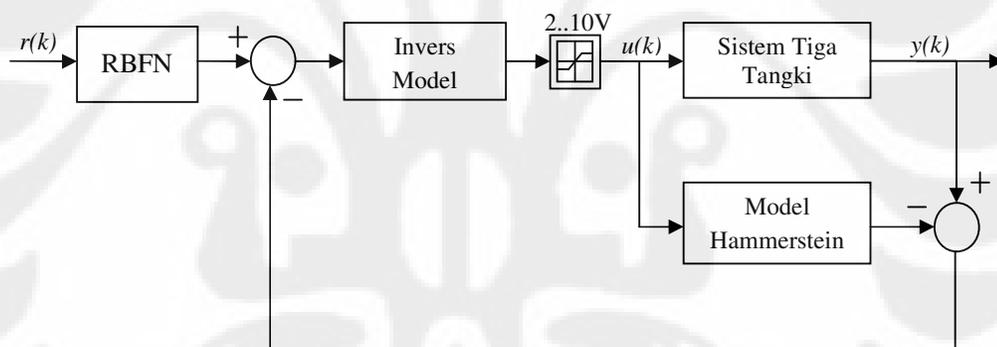
Tabel 4.1 Titik tengah dan lebar *cluster* RBFN

Cluster ke-	Titik Tengah (c_i)	Lebar <i>Cluster</i> (σ_i)
1	0.5282	0.6743
2	1.5397	0.6743
3	2.6191	0.7196
4	3.7715	0.7186
5	4.8494	0.7186
6	5.9790	0.7531
7	7.1286	0.7324
8	8.2272	0.7324
9	9.3859	0.7544

10	10.5175	0.7544
11	11.6927	0.7835
12	12.9275	0.6896
13	13.9618	0.6896

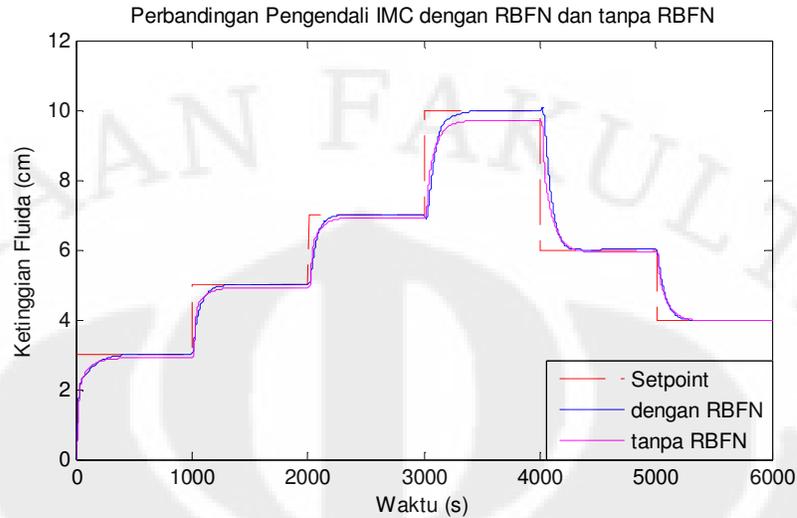
Jumlah cluster yang dihasilkan sebanyak 13 buah cukup untuk membuat RBFN memperbaiki kesalahan model yang terjadi. Penalaan yang dilakukan terhadap RBFN adalah pada nilai *learning rate* α yang digunakan. Sehingga walaupun dengan jumlah cluster yang berbeda tetap dapat memperbaiki kinerja dari RBFN dengan sama baiknya apabila dilakukan penalaan α dengan baik.

Gambar 4.5 menunjukkan diagram blok yang digunakan dalam simulasi sistem dengan pengendali IMC yang telah ditambah dengan blok RBFN yang dimaksudkan untuk memperbaiki kinerja dari pengendali IMC.



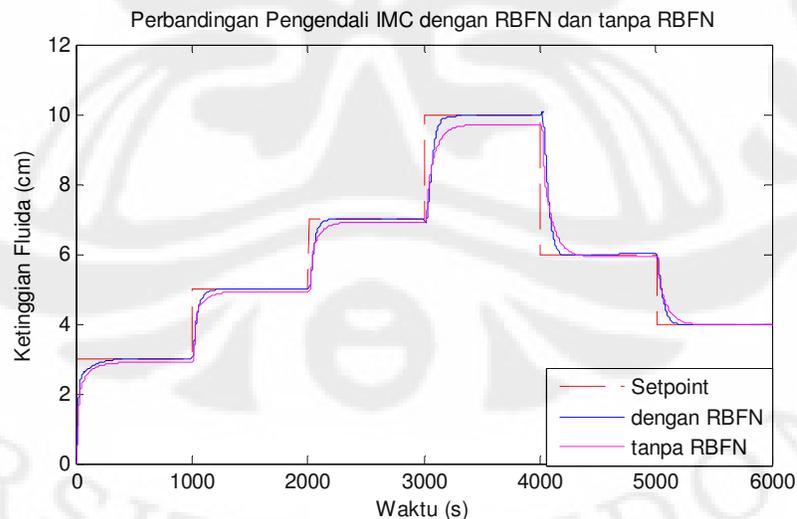
Gambar 4.5 Blok diagram simulasi pengendali IMC dengan RBFN

Hubungan masukan dan keluaran RBFN seperti pada persamaan (2.24), maka hasil dari simulasi pengendali IMC dengan RBFN pada sistem tiga tangki terhubung dapat dilihat pada gambar 4.6 sampai 4.9. Simulasi dilakukan terhadap beberapa nilai α pada rentang [0.002, 0.0045] untuk melihat pengaruh RBFN terhadap performa pengendalian dari sistem.



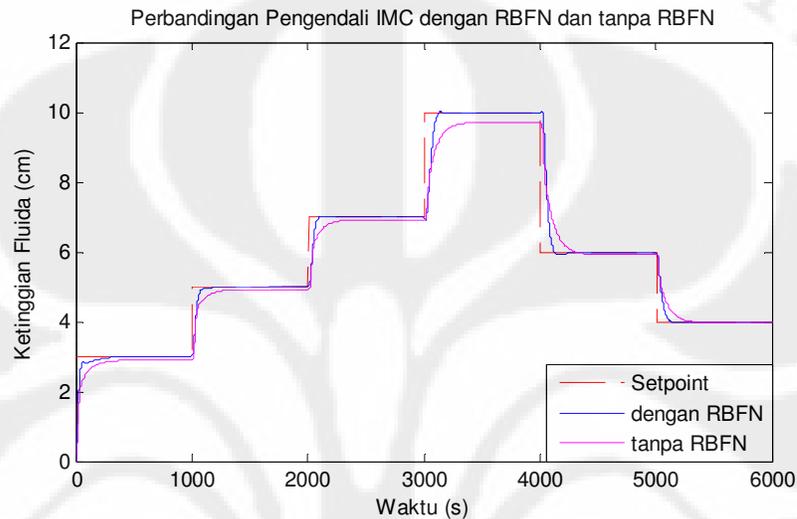
Gambar 4.6 Hasil simulasi pengendali IMC dengan RBFN dan tanpa RBFN ($\alpha = 0.002$)

Dengan menggunakan α sebesar 0.002, terlihat pada gambar 4.6 keluaran dari proses dapat mengikuti nilai acuan yang diberikan tanpa terjadi kesalahan galat tunak. Hal ini disebabkan karena parameter bobot w yang diperbarui tiap waktu cuplik seperti pada persamaan (2.16). Sehingga pada keadaan tunaknya, kesalahan yang terjadi adalah nol. Dari sisi respon alihnya, masih diperoleh *settling time* yang tidak jauh berbeda dengan pengendali tanpa RBFN untuk tiap nilai acuan.



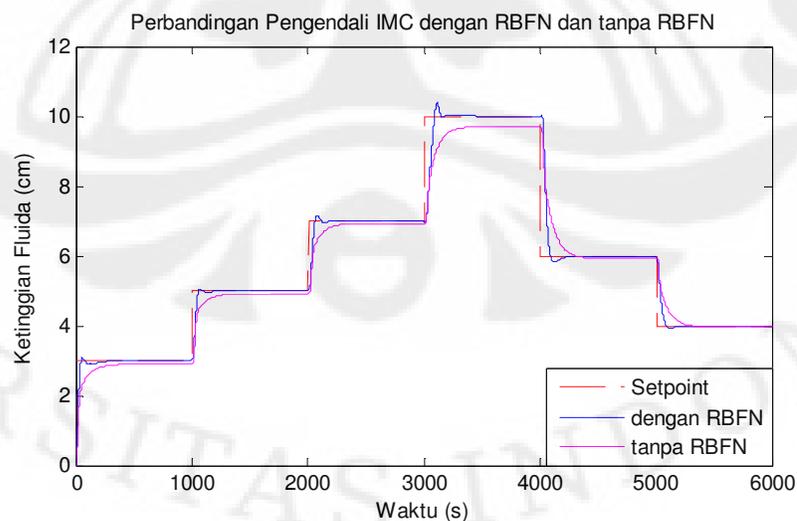
Gambar 4.7 Hasil simulasi pengendali IMC dengan RBFN dan tanpa RBFN ($\alpha = 0.0027$)

Pada gambar 4.7 dapat dilihat respon alih dari sistem dengan $\alpha = 0.0027$ memiliki *settling time* yang lebih cepat dibandingkan nilai α sebelumnya. Hal ini dikarenakan semakin besar nilai α , maka semakin besar pula perubahan parameter bobot w tiap waktu cupliknya. Dengan demikian perubahan keluaran dari sistem juga menjadi lebih besar tiap waktu cupliknya.



Gambar 4.8 Hasil simulasi pengendali IMC dengan RBFN dan tanpa RBFN ($\alpha = 0.0035$)

Dengan menggunakan α sebesar 0.0035, dapat dilihat pada gambar 4.8 bahwa respon alih dari sistem menjadi lebih baik dibandingkan dengan nilai α sebelumnya. Selain itu kesalahan galat tunak yang terjadi juga bernilai nol. Dengan demikian pengendali dapat dikatakan bekerja dengan baik.

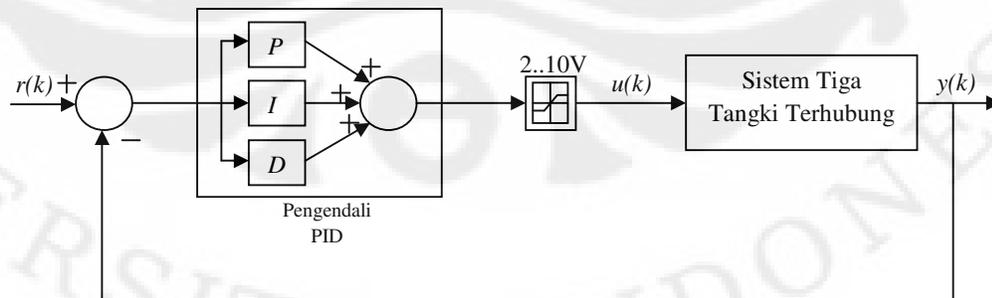


Gambar 4.9 Hasil simulasi pengendali IMC dengan RBFN dan tanpa RBFN ($\alpha = 0.0045$)

Dengan menggunakan α sebesar 0.0045, pada gambar 4.9 terlihat mulai timbul *overshoot* pada keluaran sistem, yang disebabkan perubahan parameter bobot terlalu cepat untuk tiap waktu cupliknya. Selain itu, perubahan pada respon alih dari sistem sudah tidak terlalu signifikan bila dibandingkan dengan nilai α yang sebelumnya. Oleh karena itu nilai α yang digunakan pada simulasi selanjutnya adalah $\alpha = 0.0035$. Dari keempat hasil simulasi diatas, dapat dikatakan bahwa dengan menggunakan parameter yang tepat, penambahan RBFN pada skema pengendali IMC dapat memperbaiki kinerja dari IMC, yaitu menghilangkan kesalahan galat tunak, mempercepat respons dari sistem dan juga mengkompensasi karakteristik model dan invers model yang tidak sempurna.

4.3 Perbandingan Kinerja Pengendali IMC dengan Skema Pengendali PID

Pembahasan dilanjutkan dengan melihat perbandingan kinerja antara pengendali IMC yang telah diperoleh dengan skema pengendali PID. Skema pengendali PID terdiri dari tiga bagian utama. Bagian pertama yaitu *proportional gain* yaitu bagian penguat dengan skala tertentu yang disebut K_p . Bagian yang kedua adalah bagian integral, bagian ini melakukan proses integral terhadap masukannya dan dengan adanya bagian ini, maka kesalahan galat tunak dari sistem bisa menjadi nol, penguat pada bagian ini disebut K_i . Terakhir adalah bagian yang melakukan penurunan terhadap masukannya, *derivative*, bagian ini mempengaruhi respon alih dari sistem, penguat pada bagian ini disebut K_d . Masukan ketiga bagian adalah kesalahan yang terjadi pada sistem $e(t)$ dan keluaran dari pengendali adalah penjumlahan dari keluaran ketiga bagian tersebut.



Gambar 4.10 Blok diagram simulasi sistem dengan pengendali PID

Gambar 4.8 menunjukkan blok diagram yang digunakan dalam simulasi untuk skema pengendali PID. Sebelum simulasi dilakukan, perlu dicari terlebih dahulu penguat pada masing-masing bagian pengendali (K_p , K_i dan K_d) yang optimal untuk sistem atau disebut dengan penalaan PID. Metode penalaan yang digunakan pada tulisan ini adalah metode *Ziegler-Nichols*. Dengan menggunakan metode ini, terlebih dahulu dicari penguat kritis dari sistem (K_u), yaitu penguat yang membuat keluaran sistem menjadi berosilasi dengan amplitudo yang konstan. Dengan diperolehnya K_u , dapat diperoleh juga nilai periode dari osilasi keluaran, T_u . Setelah nilai K_p , K_i dan K_d dapat diperoleh menurut karakteristik pada tabel 4.2.

Tabel 4.2 Karakteristik penalaan PID dengan metode *Ziegler-Nichols*

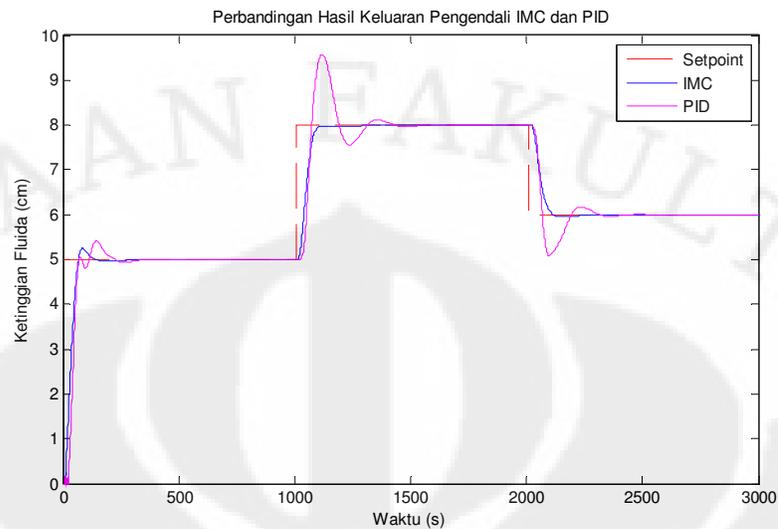
Jenis Pengendali	<i>Proportional Gain</i> (K_p)	<i>Integral Time</i> (T_i)	<i>Derivative Time</i> (T_d)
P	$K_u/2$	-	-
PI	$K_u/2.2$	$T_u/2$	-
PID	$K_u/1.7$	$T_u/2$	$T_u/8$

Dengan hubungan T_i dan T_d dengan K_i dan K_d adalah

$$T_i = \frac{K_c}{K_i} \quad (4.1)$$

$$T_d = \frac{K_d}{K_c} \quad (4.2)$$

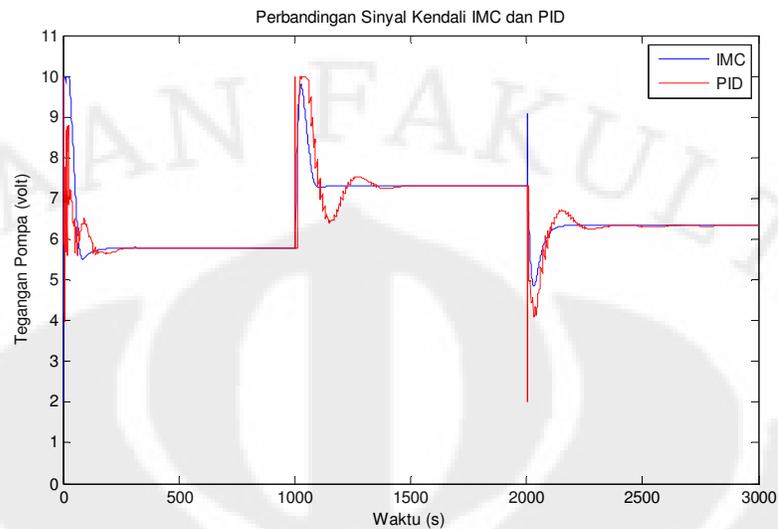
Untuk sistem tiga tangki ini diperoleh K_u dan T_u masing-masing sebesar 2 dan 90 detik. Kemudian dengan menggunakan tabel 4.1, persamaan (4.1) dan (4.2), maka dapat diperoleh nilai K_p , K_i dan K_d masing-masing sebesar 1.1765, 0.0261 dan 13.2353. Setelah diperoleh nilai parameter-parameter tersebut, maka sistem tiga tangki terbungkus dengan pengendali PID dapat disimulasikan. Grafik perbandingan nilai acuan dan keluaran sistem hasil simulasi dengan pengendali IMC dan PID dapat dilihat pada gambar 4.11.



Gambar 4.11 Hasil simulasi perbandingan pengendali IMC dan PID

Pada gambar 4.11 terlihat performa dari pengendali IMC lebih baik dibandingkan dengan pengendali PID pada sistem tiga tangki terhubung ini. Walaupun keduanya memiliki kesalahan galat tunak yang sama-sama bernilai nol, tetapi pengendali PID memiliki respons alih yang buruk. Hal ini dapat dilihat dari besarnya persentase *overshoot* yang terjadi, yaitu mencapai 19% pada nilai acuan ketinggian fluida 8 cm, sedangkan pada pengendali IMC hanya 4% pada nilai acuan 5 cm. Selain itu, pengendali PID juga menunjukkan *settling time* yang lebih lama, yaitu 255 detik, dibandingkan dengan *settling time* yang dihasilkan pengendali IMC, yaitu 68 detik pada nilai acuan ketinggian fluida 8 cm.

Pada gambar 4.12 dapat dilihat sinyal kendali yang dihasilkan dari kedua jenis pengendali konvergen pada nilai yang sama, yaitu nilai yang membuat kesalahan galat tunak bernilai nol. Namun untuk mencapai nilai tersebut, sinyal kendali dari pengendali IMC konvergen lebih cepat dibandingkan dengan sinyal kendali dari pengendali PID. Hal inilah yang membuat respons alih dari pengendali IMC lebih baik dibandingkan dengan PID.

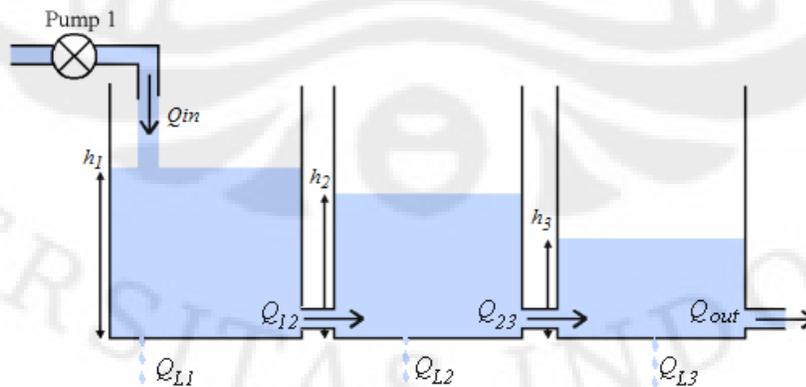


Gambar 4.12 Perbandingan sinyal kendali IMC dan PID

Dari kedua perbandingan diatas, dapat disimpulkan bahwa pengendali IMC memiliki kinerja yang lebih baik dibandingkan dengan pengendali PID untuk sistem tiga tangki terhubung.

4.4 Kinerja Pengendali IMC Terhadap Gangguan

Untuk melihat kinerja sistem tiga tangki terhubung ketika terjadi gangguan, maka berikutnya disimulasikan terjadinya kebocoran pada sistem. Kebocoran ini terjadi pada tiap tangki dan dianggap sebagai gangguan pada sistem yang akan mengubah fungsi alih dari persamaan model matematis sistem tiga tangki yang digunakan.



Gambar 4.13 Sistem tiga tangki terhubung dengan kebocoran

$$\begin{aligned}\frac{A_1 dh_1}{dt} &= Q_{in} - Q_{12} - Q_{L1} \\ &= Q_{in} - a_1 \operatorname{sgn}(h_1 - h_2) \sqrt{|h_1 - h_2|} - S_{L1} \sqrt{2gh_1}\end{aligned}\quad (4.3)$$

$$\begin{aligned}\frac{A_2 dh_2}{dt} &= Q_{12} - Q_{23} - Q_{L2} \\ &= a_1 \operatorname{sgn}(h_1 - h_2) \sqrt{|h_1 - h_2|} - a_2 \operatorname{sgn}(h_2 - h_3) \sqrt{|h_2 - h_3|} - S_{L2} \sqrt{2gh_2}\end{aligned}\quad (4.4)$$

$$\begin{aligned}\frac{A_3 dh_3}{dt} &= Q_{23} - Q_{out} - Q_{L3} \\ &= a_2 \operatorname{sgn}(h_2 - h_3) \sqrt{|h_2 - h_3|} - a_3 \sqrt{h_3} - S_{L3} \sqrt{2gh_3}\end{aligned}\quad (4.5)$$

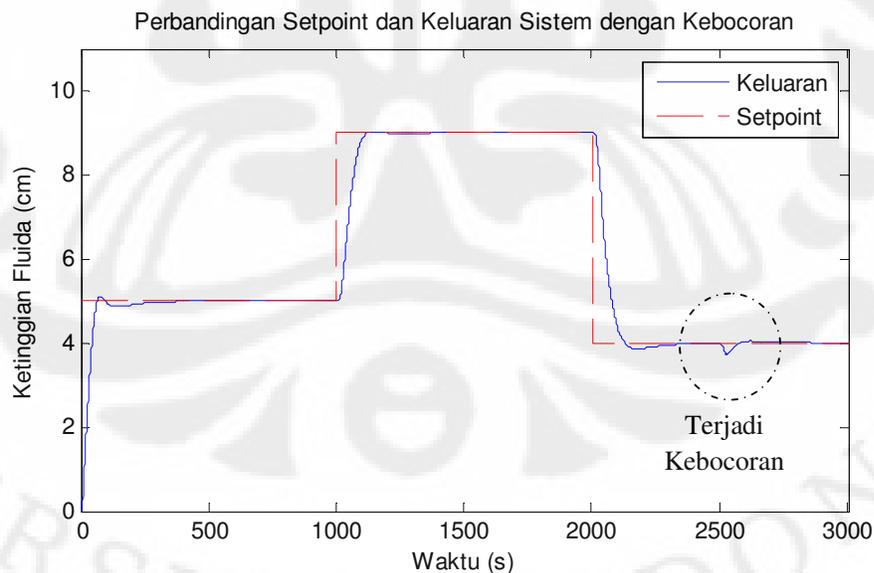
dengan

S_{L1} = luas kebocoran pada tangki 1 (cm^2)

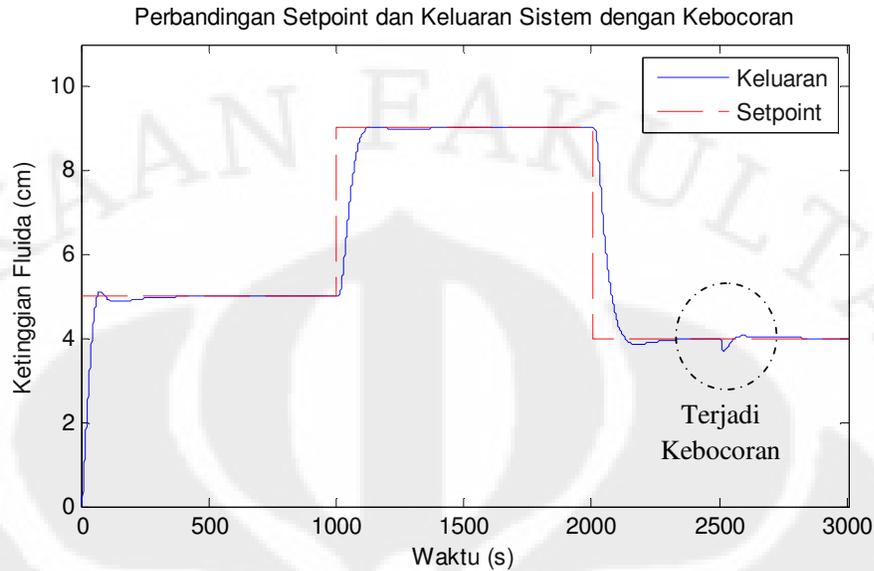
S_{L2} = luas kebocoran pada tangki 2 (cm^2)

S_{L3} = luas kebocoran pada tangki 3 (cm^2)

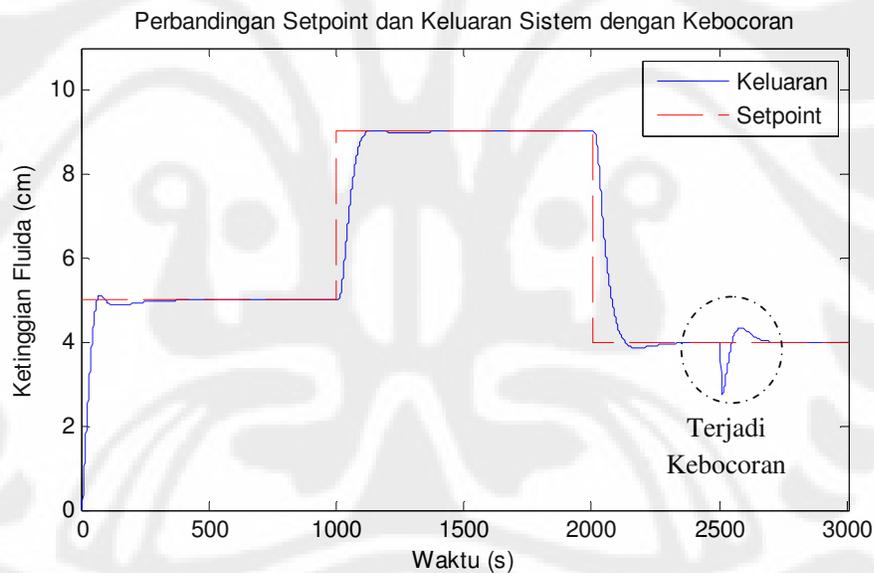
Gambar 4.13 menunjukkan sistem tiga tangki terhubung dengan kebocoran yang digunakan dalam simulasi. Kebocoran pada tangki yang disimulasikan memiliki luas 0.1 cm^2 dan terjadi pada detik ke 2500.



Gambar 4.14 Respons sistem terhadap kebocoran pada tangki 1.



Gambar 4.15 Respons sistem terhadap kebocoran pada tangki 2.

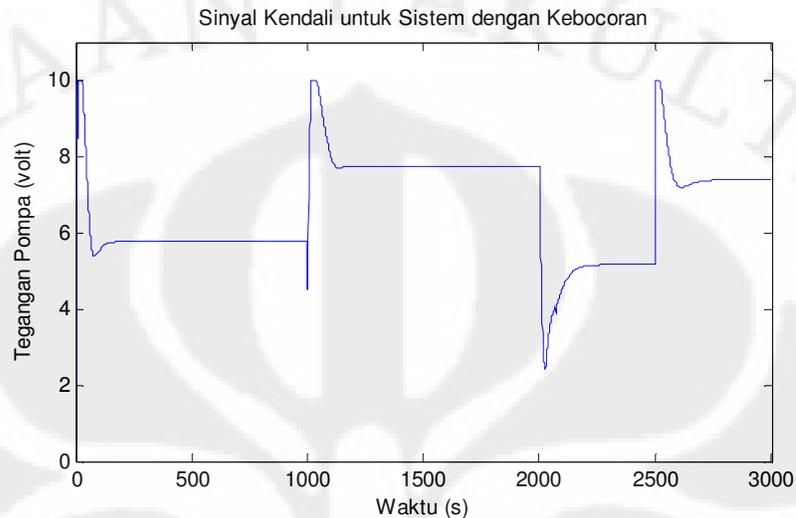


Gambar 4.16 Respons sistem terhadap kebocoran pada tangki 3

Gambar 4.14, 4.15 dan 4.16 menunjukkan respons sistem terhadap kebocoran di tangki pertama, kedua dan ketiga. Pada detik ke 2500, ketika kebocoran terjadi, grafik menunjukkan adanya gangguan pada keluaran dari sistem. Walaupun demikian pengendali IMC dapat mengembalikan keluaran sistem ke nilai acuannya dalam waktu yang cukup cepat. Besarnya penyimpangan bergantung pada besarnya kebocoran yang terjadi dan dimana kebocoran terjadi. Semakin besar kebocorannya, semakin besar juga penyimpangannya.

Universitas Indonesia

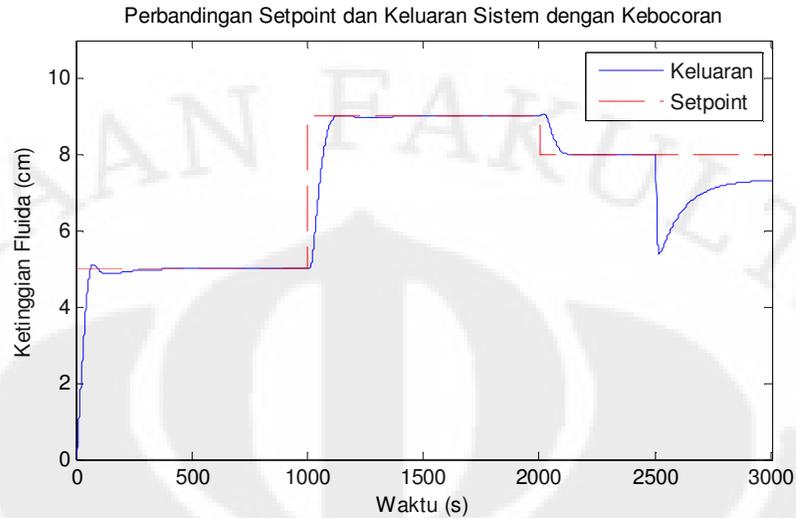
Penyimpangan terbesar terjadi ketika kebocoran ada di tangki ke tiga, karena kebocoran tersebut secara langsung mempengaruhi keluaran dari sistem.



Gambar 4.17 Sinyal kendali untuk sistem dengan kebocoran pada tangki 3

Gambar 4.17 menunjukkan grafik sinyal kendali yang dihasilkan untuk sistem dengan kebocoran pada tangki 3. Ketika kebocoran terjadi, maka sinyal kendali yang dihasilkan oleh pengendali IMC menyesuaikan keluaran agar menjadi sama dengan nilai acuannya. Hal ini dapat dilihat dari meningkatnya nilai tunak dari pengendali setelah detik ke 2500, yaitu dari 5.17 V menjadi 7.37 V saat kebocoran terjadi.

Terjadinya kebocoran juga mempengaruhi ketinggian fluida maksimum yang dapat dicapai sistem. Sinyal kendali maksimum yang dapat diberikan pengendali ke sistem tiga tangki terhubung adalah 10 V untuk ketinggian fluida pada kondisi tunak 15 cm. Dengan adanya kebocoran maka terjadi penurunan tinggi maksimum fluida yang dapat dicapai. Sehingga bila sistem diberikan nilai acuan melebihi tinggi maksimum tersebut, maka pengendali IMC tidak dapat membuat keluaran sistem menuju nilai acuan yang dikarenakan terbatasnya sinyal kendali yang dapat diberikan ke sistem.



Gambar 4.18 Respons sistem untuk nilai acuan melebihi tinggi maksimum

Pada gambar 4.18 terlihat bahwa pengendali sudah tidak mampu untuk mencapai nilai acuan yang diberikan. Pada kondisi tangki 3 bocor sebesar 0.15 cm^2 , pengendali hanya dapat mengendalikan sistem dengan ketinggian fluida maksimum 7.35 cm . Ketinggian maksimum ini dipengaruhi oleh besarnya kebocoran yang terjadi dan dimana kebocoran tersebut terjadi.

BAB 5

KESIMPULAN

Dari hasil simulasi identifikasi dan pengendalian dengan menggunakan skema IMC pada sistem tiga tangki terhubung, maka dapat disimpulkan bahwa:

1. Hasil identifikasi dengan menggunakan model Hammerstein dengan neural network untuk bagian nonlinear dan estimator ARX untuk bagian linearnya dapat mewakili proses nonlinear yang terjadi pada sistem tiga tangki terhubung dengan nilai RMSE = 0.2258 dan VAF = 98.69%.
2. Pengendali dengan skema Internal Model Control (IMC) dapat mengendalikan sistem dengan baik bila model dan inverse model dapat meniru proses dan kebalikan dari proses.
3. Penambahan RBFN secara cascade pada pengendali IMC dapat menghilangkan kesalahan galat tunak yang terjadi dan memperbaiki respons alih dari sistem.
4. Pengendali IMC memiliki kinerja yang lebih baik dibandingkan dengan skema pengendali PID untuk sistem tiga tangki terhubung.
5. Pengendali IMC dapat mengatasi adanya kebocoran pada tangki, namun memiliki keterbatasan yang bergantung dari besarnya luas kebocoran yang terjadi dan letak kebocoran tersebut.

DAFTAR ACUAN

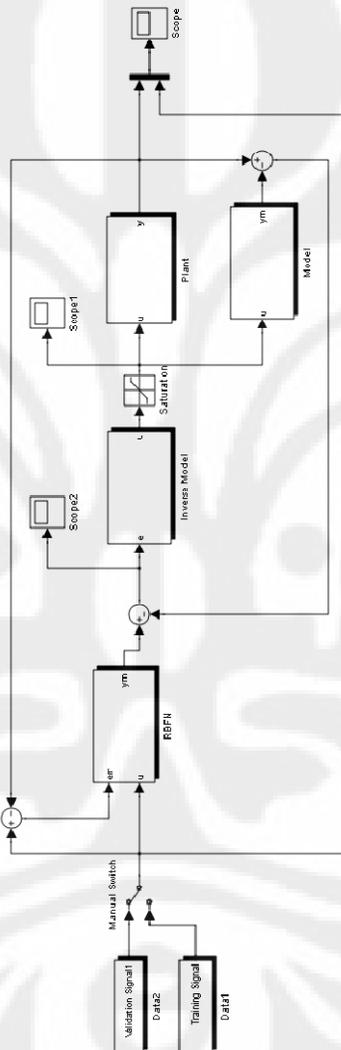
- [1] Zhicheng, Zhao. *A New-model Internal Model Control Scheme Based on Neural Network*". IEEE, 2008.
- [2] Johansson, Rolf. *System Modelling and Identification*. Prentice Hall International Edition, 1993.
- [3] Li, Jia. *Neuro-fuzzy System Based Identification Method for Hammerstein Processes*. IEEE, 2004.
- [4] Vieira, Jose. *Parameter Estimation of Nonlinear Systems with Hammerstein Models Using Neurofuzzy and Polynomial Approximation Approaches*. IEEE, 2004.
- [5] Liutkevičius. *Fuzzy Hammerstein Model of Nonlinear Plant*. Department of Applied Informatics, Vytautas Magnus University, 2008.
- [6] Subiantoro, Aries. *Sistem Kendali Adaptif*. Diktat Kuliah, 2002.
- [7] Gupta, Madan. *Static and Dynamic Neural Networks*. John Wiley and Sons, Inc. 2003.
- [8] Maraden, Yan. *"Identifikasi Sistem Tiga Tangki Terhubung dengan Menggunakan Algoritma Fuzzy Clustering Gustafson-Kessel*. Departemen Elektro Fakultas Teknik Universitas Indonesia, 2004.

DAFTAR PUSTAKA

1. Babuska, Robert. *Fuzzy Modelling for Control*. Kluwer Academic Publisher, 1998.
2. Gupta, Madan. *Static and Dynamic Neural Networks*. John Wiley and Sons, Inc. 2003.
3. Johansson, Rolf. *System Modelling and Identification*. Prentice Hall International Edition, 1993.
4. Ross, Timothy. *Fuzzy Logic with Engineering Applications*. McGraw-Hill, Inc., 1995.
5. Maraden, Yan. *Identifikasi Sistem Tiga Tangki Terhubung dengan Menggunakan Algoritma Fuzzy Clustering Gustafson-Kessel*. Departemen Elektro Fakultas Teknik Universitas Indonesia, 2004.
6. Tovanno, Fernando. *Perancangan Sistem Kendali Nonlinear dengan Menggunakan Mode Fuzzy Takagi-Sugeno pada Sistem Tangki Terhubung*. Departemen Elektro Fakultas Teknik Universitas Indonesia, 2005.

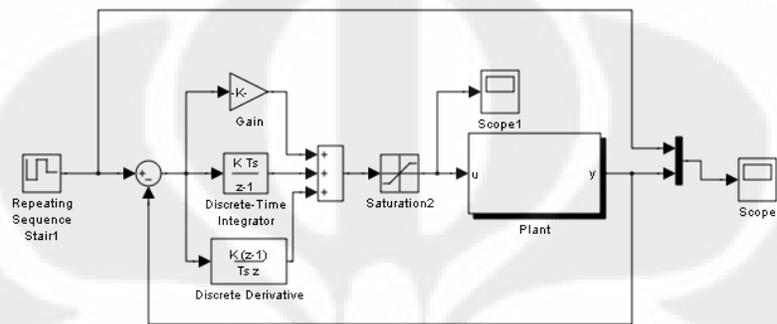
LAMPIRAN A

Gambar simulasi pengendali IMC untuk sistem tiga tangki terhubung



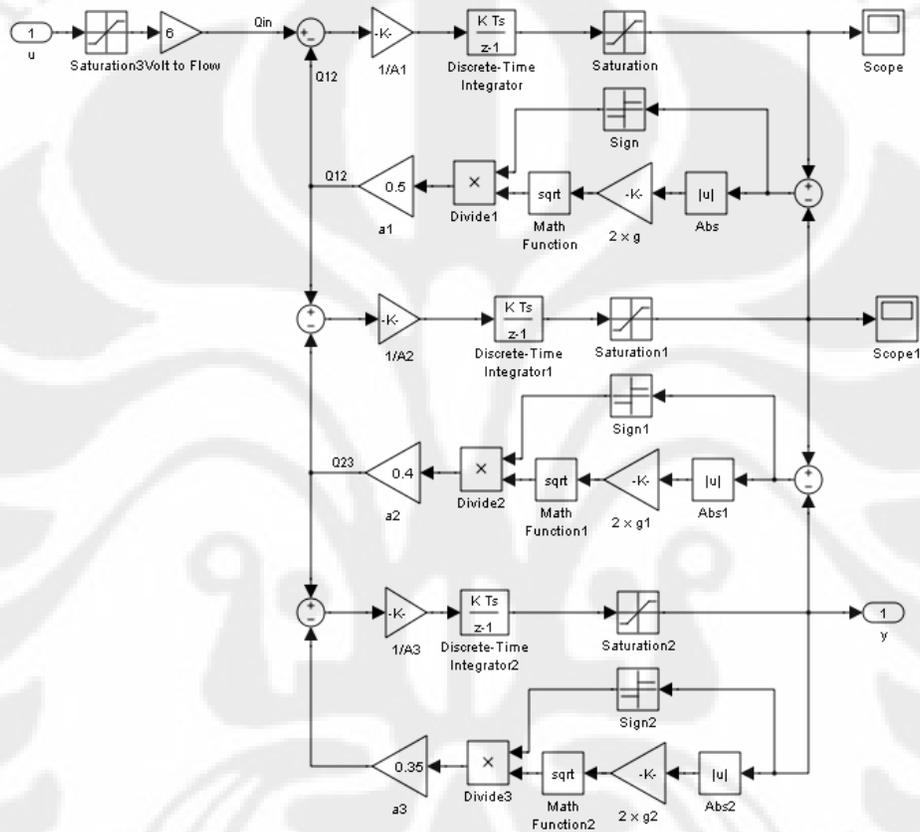
LAMPIRAN B

Gambar simulasi pengendali PID untuk sistem tiga tangki terhubung



LAMPIRAN C

Gambar simulasi proses nonlinear sistem tiga tangki terhubung



LAMPIRAN D

Gambar simulasi proses nonlinear sistem tiga tangki terhubung dengan kebocoran

